MIS7420

Seminar in Management Information Systems: Paper Replication with R

Prof. Zhiqiang Zheng, Prof. Vijay Mookerjee

Author: Beyza Celik, Luoying Chen, Duc Vu, Yihong Liu
NETID: BXC190000, LXC190027, DDV110020, YXL180111

Contents

List of Figures

List of Tables

List of Codes

1 Data Cleaning Process

In this section, we present our codes for data cleaning and panel data preparation. Package `dplyr` Wickham et al. (2015), `haven` Wickham and Miller (2018), `sqldf` Grothendieck (2017), `zoo` Zeileis and Grothendieck (2005), `plm` Croissant et al. (2008), are used in this process.

Notice that the provided data only contain users' browsing transactions with purchase. When we apply search engine filter on reference domain, besides google.com, yahoo.com, bing.com, some other search engines (msn.com, aol.com, live.com, mywebsearch.com) and five vendors (amazon.com, staples.com, dell.com, walmart.com, bestbuy.com) are also involved. For product categories, we only consider those sold at Circuit City and exclude three other types (Business machines, Office furniture, Office supplies).

For `CCStorePresent`, its value is the same as `Store_Close_Status` in original data. And we set `AfterStoreClosing` to 0 if the time of user transaction is before November, 2008 and 1 otherwise. For `BBStorePresent`, we set it as 1 for user transactions if there's any Best Buy stores around user's location and 0 otherwise, using the `bestbuyzipcodes_sample` dataset. `NoReferringDomain` is the variable we construct if the user directly came to the target website without any referring domain. And `ReferringDomainIsSearchEngine` is the variable we construct if the user was referred by search engine to the target website.

Finally, two panel data are constructed by concatenating `sales_allotherzipcode` with `sales_ccity0mile` and `sales_ccity5mile`. After the concatenation and aggregation, we found that the built panel data are unbalanced, in a sense that, for instance, `zip_code` 75080 only has 2 records, instead of 24 (2 years). It happens because (a) the provided data are a small sample from the whole original one; (b) the original data might not cover the full 2 years period. Unbalanced panel data has been studied by many researchers Baltagi and Song (2006), like unbalanced seemingly unrelated regression McDowell (2004). Here we adopt a naive solution: we impute the missing values for one `zip_code` and one target domain by averaging those non-missing values of this `zip_code`.

```r
# load library
library('dplyr')
library('haven')
library('sqldf')
library('zoo')
library('plm')
library('stargazer')

# all data path
bb_zipcode_path <- 'data/bestbuyzipcodes_sample.sas7bdat'
sales_allother_zipcode_path <- 'data/sales_allotherzipcode_sample.sas7bdat'
sales_cc_0mile_path <- 'data/sales_ccity0milezipcode_sample.sas7bdat'
sales_cc_5miles_path <- 'data/sales_ccity5milezipcode_sample.sas7bdat'

# load data
bb_zipcode <- read_sas(bb_zipcode_path)
sales_allother_zipcode <- read_sas(sales_allother_zipcode_path)
sales_cc_0mile <- read_sas(sales_cc_0mile_path)
sales_cc_5miles <- read_sas(sales_cc_5miles_path)

# Data Mapping
sales_allother_zipcode$Store_Close_Status <- 0 # NaN means no CC in 5-miles radius, we change NaN to 0

# Exclude Data without purchase
# All data should be with purchase -> tran_flg == 1
```

```
sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$tran_flg == 1,]
sales_cc_0mile <- sales_cc_0mile[sales_cc_0mile$tran_flg == 1,]
sales_cc_5miles <- sales_cc_5miles[sales_cc_5miles$tran_flg == 1,]

# Filter Referring Domain

# groupby ref_domain and count
groupby_ref_domain_result <- aggregate(machine_id ~ ref_domain_name, rbind(sales_allother_zipcode, sales_cc_0mile, sales_cc_5miles), FUN = "length"
      )
groupby_ref_domain_result <- groupby_ref_domain_result[order(-groupby_ref_domain_result$machine_id), ]
# we identify some search engines
search_engine_to_consider1 <- c("GOOGLE.COM", "YAHOO.COM", "google.com", "yahoo.com",
                             "MSN.COM", "msn.com", "aol.com", "AOL.COM", "LIVE.COM", "live.com",
                             "MYWEBSEARCH.COM", "ASK.COM", "MYWAY.COM", "mywebsearch.com",
                             "ask.com", "YAHOO.NET", "BIZRATE.COM", "bizrate.com",
                             "amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com",
                             "AMAZON.COM", "STAPLES.COM", "DELL.COM", "WALMART.COM", "BESTBUY.COM")

search_engine_to_consider2 <- c("GOOGLE.COM", "YAHOO.COM", "BING.COM", "google.com", "yahoo.com", "bing.com")

ref_domain_to_consider1 <- c("", "GOOGLE.COM", "YAHOO.COM", "google.com", "yahoo.com",
                             "MSN.COM", "msn.com", "aol.com", "AOL.COM", "LIVE.COM", "live.com",
                             "MYWEBSEARCH.COM", "ASK.COM", "MYWAY.COM", "mywebsearch.com",
                             "ask.com", "YAHOO.NET", "BIZRATE.COM", "bizrate.com",
                             "amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com",
                             "AMAZON.COM", "STAPLES.COM", "DELL.COM", "WALMART.COM", "BESTBUY.COM")

ref_domain_to_consider2 <- c("", "GOOGLE.COM", "YAHOO.COM", "BING.COM", "google.com", "yahoo.com", "bing.com")

# Then we filter data by refer domain name
sales_allother_zipcode <- sales_allother_zipcode[(sales_allother_zipcode$ref_domain_name %in% ref_domain_to_consider1),]
sales_cc_0mile <- sales_cc_0mile[(sales_cc_0mile$ref_domain_name %in% ref_domain_to_consider1),]
sales_cc_5miles <- sales_cc_5miles[(sales_cc_5miles$ref_domain_name %in% ref_domain_to_consider1),]

# Filter Target Domain Name
groupby_target_domain_result <- aggregate(machine_id ~ domain_name, rbind(sales_allother_zipcode, sales_cc_5miles), FUN = "length")
groupby_target_domain_result <- groupby_target_domain_result[order(-groupby_target_domain_result$machine_id), ]
five_target_domain_to_consider <- c("amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com")
two_target_domain_to_consider <- c("amazon.com","bestbuy.com")

# we can choose what filter to apply
sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$domain_name %in% five_target_domain_to_consider,]
sales_cc_0mile <- sales_cc_0mile[sales_cc_0mile$domain_name %in% five_target_domain_to_consider,]
sales_cc_5miles <- sales_cc_5miles[sales_cc_5miles$domain_name %in% five_target_domain_to_consider,]

# Product Categories
# 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
# Jay removed 28, 30, 39, 40
# We choose to remove 38 39 40
sort(unique(rbind(sales_allother_zipcode, sales_cc_0mile, sales_cc_5miles)$prod_category_id))
category_to_consider <- c(22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37)
experience_product <- c(24, 25, 26, 27, 28, 31, 32, 33, 34, 36, 37)
search_product <- c(22, 23, 24, 29, 30, 35)

sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$prod_category_id %in% category_to_consider,]
sales_cc_0mile <- sales_cc_0mile[sales_cc_0mile$prod_category_id %in% category_to_consider,]
sales_cc_5miles <- sales_cc_5miles[sales_cc_5miles$prod_category_id %in% category_to_consider,]

# Date Transform
sales_allother_zipcode$event_date <- as.Date(sales_allother_zipcode$event_date)
sales_cc_0mile$event_date <- as.Date(sales_cc_0mile$event_date)
sales_cc_5miles$event_date <- as.Date(sales_cc_5miles$event_date)

# construct MonthYear - month of year
sales_allother_zipcode$MonthYear <- format(sales_allother_zipcode$event_date, "%Y-%m")
sales_cc_0mile$MonthYear <- format(sales_cc_0mile$event_date, "%Y-%m")
sales_cc_5miles$MonthYear <- format(sales_cc_5miles$event_date, "%Y-%m")

# Mark CC Closure

# CCStorePresent
# it is the same as Store_Close_Status
```

```
sales_allother_zipcode$CCStorePresent <- sales_allother_zipcode$Store_Close_Status
sales_cc_0mile$CCStorePresent <- sales_cc_0mile$Store_Close_Status
sales_cc_5miles$CCStorePresent <- sales_cc_5miles$Store_Close_Status

# AfterStoreClosing
sales_allother_zipcode$AfterStoreClosing <- ifelse(sales_allother_zipcode$MonthYear < "2008-11", 0, 1)
sales_cc_0mile$AfterStoreClosing <- ifelse(sales_cc_0mile$MonthYear < "2008-11", 0, 1)
sales_cc_5miles$AfterStoreClosing <- ifelse(sales_cc_5miles$MonthYear < "2008-11", 0, 1)

# BBStorePresent
sales_allother_zipcode <- merge(sales_allother_zipcode, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)
sales_cc_0mile <- merge(sales_cc_0mile, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)
sales_cc_5miles <- merge(sales_cc_5miles, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)

sales_allother_zipcode$BBStorePresent <- na.fill(sales_allother_zipcode$BB_Store_Status, 0)
sales_cc_0mile$BBStorePresent <- na.fill(sales_cc_0mile$BB_Store_Status, 0)
sales_cc_5miles$BBStorePresent <- na.fill(sales_cc_5miles$BB_Store_Status, 0)

# Mark Referring Domain
# Question: How to group data?
sales_allother_zipcode$NoReferringDomain <- ifelse(sales_allother_zipcode$ref_domain_name == "", 1, 0)
sales_cc_0mile$NoReferringDomain <- ifelse(sales_cc_0mile$ref_domain_name == "", 1, 0)
sales_cc_5miles$NoReferringDomain <- ifelse(sales_cc_5miles$ref_domain_name == "", 1, 0)

sales_allother_zipcode$ReferringDomainIsSearchEngine <- ifelse(sales_allother_zipcode$ref_domain_name %in% search_engine_to_consider1, 1, 0)
sales_cc_0mile$ReferringDomainIsSearchEngine <- ifelse(sales_cc_0mile$ref_domain_name %in% search_engine_to_consider1, 1, 0)
sales_cc_5miles$ReferringDomainIsSearchEngine <- ifelse(sales_cc_5miles$ref_domain_name %in% search_engine_to_consider1, 1, 0)

# Aggregate Data
concat_data1 <- rbind(sales_allother_zipcode, sales_cc_0mile)
concat_data2 <- rbind(sales_allother_zipcode, sales_cc_5miles)
concat_data1_exp <- concat_data1[concat_data1$prod_category_id %in% experience_product, ]
concat_data1_search <- concat_data1[concat_data1$prod_category_id %in% search_product, ]
concat_data2_exp <- concat_data2[concat_data2$prod_category_id %in% experience_product, ]
concat_data2_search <- concat_data2[concat_data2$prod_category_id %in% search_product, ]
```

Code 1: Data Preprocess

2 Paper Replication

In this section, we provide our replication for this paper. Names for subsections correspond to the tables in the published paper. Package `stargazer` Hlavac (2015) and `estout` Jann (2004) are used to export estimation into LaTeX format.

2.1 Table 1

Table 1 shows the summary statistics of top five vendors by sales volume.

Table 1: Summary Statistics of Top Five Vendors by Sales Volume

Domain Name	**Total Transactions**	**Total Sales**	**Total Pages Viewed**	**Pages Per Dollar**	**Total Duration**	**Mins Per Dollar**
dell.com	1,620	483,703.300	66,953	0.138	57,225.660	0.118
amazon.com	10,904	354,573.300	464,383	1.310	369,227.900	1.041
staples.com	5,927	236,982.300	247,163	1.043	166,189.900	0.701
walmart.com	1,977	156,606.100	80,397	0.513	68,434.890	0.437
bestbuy.com	1,230	149,950.400	50,627	0.338	36,735.900	0.245

Codes for generating Table 1 are listed below.

```
# Table 1
table1_raw <-  rbind(read_sas(sales_allother_zipcode_path), read_sas(sales_cc_0mile_path))
table1 <- sqldf("SELECT domain_name as DomainName, count(*) as TotalTransaction, SUM(prod_totprice) AS TotalSales, SUM(pages_viewed) AS
      TotalPagesViewed, SUM(pages_viewed)/SUM(prod_totprice) AS PagesPerDollar, SUM(duration) AS TotalDuration, SUM(duration)/SUM(prod_totprice) AS
       MinsPerDollar FROM table1_raw GROUP BY domain_name ORDER BY TotalSales DESC")
stargazer(table1[1:5,], align=TRUE, summary = FALSE, rownames = FALSE, title="Summary Statistics of Top Five Vendors by Sales Volume")
```

Code 2: Table 1 Generation

2.2 Table 2

Table 2 summarizes the frequency of referral channels for various online retailers.

Table 2: Summary Statistics of Referring Domain Categories

Domain Name	**Total Transactions**	**Referred by Search Engine**	**Direct to Website**	**Referred by Others**
amazon.com	10,904	2,955(27.1%)	7,018(64.4%)	931(8.6%)
bestbuy.com	1,230	258(21.0%)	901(73.3%)	71(5.8%)
All Others	36,794	6,999(19.0%)	25,483(69.3%)	4,312(11.7%)
All Transactions	48,928	10,212(20.9%)	33,402(68.3%)	5,314(10.9%)

Codes for generating Table 2 are listed below.

```
# Table 2

table2_raw <-  rbind(read_sas(sales_allother_zipcode_path), read_sas(sales_cc_0mile_path))
table2_raw$direct_to_website <- ifelse(table2_raw$ref_domain_name == '', 1, 0)
table2_raw$referred_by_search <- ifelse(table2_raw$ref_domain_name %in% search_engine_to_consider1, 1, 0)
table2_raw$referred_by_other <- ifelse(!(table2_raw$ref_domain_name %in% ref_domain_to_consider1), 1, 0)
table2_raw$domain_name[!(table2_raw$domain_name %in% c('amazon.com', 'bestbuy.com'))] <- "All Others"
```

Code 3: Table 2 Generation

2.3 Table 3

Table 3 reports the model-free average DID values for some outcome variables.

Table 3: Average Difference-in-Difference (DID) of the Outcome Variables

Outcome Variable	**Groups**	**After Store Closure**	**Before Store Closure**	**First Difference** (se)	**DID**
Amazon Sales	Control	3.418	3.303	0.115 (0.031)	-0.167
	Treatment	3.351	3.403	-0.052 (0.212)	
Amazon PagesPerDollar	Control	1.188	1.147	0.041 (0.025)	0.257
	Treatment	1.363	1.065	0.298 (0.153)	
Amazon MinsPerDollar	Control	1.016	0.975	0.041 (0.025)	0.263
	Treatment	1.187	0.882	0.304 (0.137)	
bestbuy.com Sales	Control	3.418	3.303	0.354 (0.031)	0.623
	Treatment	3.351	3.403	0.976 (0.212)	
bestbuy.com PagesPerDollar	Control	1.188	1.147	-0.109 (0.025)	0.074
	Treatment	1.363	1.065	-0.035 (0.153)	
bestbuy.com MinsPerDollar	Control	1.016	0.975	-0.084 (0.025)	-0.012
	Treatment	1.187	0.882	-0.096 (0.137)	

Codes for generating Table 3 are listed below.

```r
# Table 3
temp <- read_sas(sales_allother_zipcode_path)
temp$Store_Close_Status <- 0
table3_0m_raw <-  rbind(temp, read_sas(sales_cc_0mile_path))
table3_5m_raw <-  rbind(temp, read_sas(sales_cc_5miles_path))

# Date Transform
table3_0m_raw$event_date <- as.Date(table3_0m_raw$event_date)
table3_5m_raw$event_date <- as.Date(table3_5m_raw$event_date)

# construct MonthYear - month of year
table3_0m_raw$MonthYear <- format(table3_0m_raw$event_date, "%Y-%m")
table3_5m_raw$MonthYear <- format(table3_5m_raw$event_date, "%Y-%m")

# Mark CC Closure

# CCStorePresent
```

```r
# it is the same as Store_Close_Status
table3_0m_raw$CCStorePresent <- table3_0m_raw$Store_Close_Status
table3_5m_raw$CCStorePresent <- table3_5m_raw$Store_Close_Status

# AfterStoreClosing
table3_0m_raw$AfterStoreClosing <- ifelse(table3_0m_raw$MonthYear < "2008-11", 0, 1)
table3_5m_raw$AfterStoreClosing <- ifelse(table3_5m_raw$MonthYear < "2008-11", 0, 1)

# BBStorePresent
table3_0m_raw <- merge(table3_0m_raw, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)
table3_5m_raw <- merge(table3_5m_raw, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)

table3_0m_raw$BBStorePresent <- na.fill(table3_0m_raw$BB_Store_Status, 0)
table3_5m_raw$BBStorePresent <- na.fill(table3_5m_raw$BB_Store_Status, 0)

# aggregate data

table3_0m_aggregate <- sqldf("SELECT Zip_Code, MonthYear, domain_name, count(*) AS TotalTransactions, SUM(pages_viewed) as TotalPages, SUM(prod_totprice) as TotalMonthlySales, SUM(duration) as TotalMins, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM table3_0m_raw GROUP BY Zip_Code, MonthYear, domain_name")
table3_5m_aggregate <- sqldf("SELECT Zip_Code, MonthYear, domain_name, count(*) AS TotalTransactions, SUM(pages_viewed) as TotalPages, SUM(prod_totprice) as TotalMonthlySales, SUM(duration) as TotalMins, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM table3_5m_raw GROUP BY Zip_Code, MonthYear, domain_name")

# Table 3 Gen Func
table3_gen <- function(table3_raw, domain_name_used, print_name){
  # Amazon Sales
  # for control
  amazonsales_control_before <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMonthlySales
  amazonsales_control_after <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalMonthlySales

  amazonsales_control_before <- log(amazonsales_control_before + 1)
  amazonsales_control_after <- log(amazonsales_control_after + 1)
  # t test
  t_test.amazonsales_control <- t.test(amazonsales_control_after, amazonsales_control_before)
  amazonsales_control_mean_diff_se <- t_test.amazonsales_control$stderr
  t_test.amazonsales_control$p.value
  amazonsales_control_after_mean <- t_test.amazonsales_control$estimate[["mean of x"]]
  amazonsales_control_before_mean <- t_test.amazonsales_control$estimate[["mean of y"]]
  amazonsales_control_mean_diff <- t_test.amazonsales_control$estimate[["mean of x"]] - t_test.amazonsales_control$estimate[["mean of y"]]

  # Amazon Sales
  # for treatment
  amazonsales_treatment_before <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMonthlySales
  amazonsales_treatment_after <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalMonthlySales

  amazonsales_treatment_before <- log(amazonsales_treatment_before + 1)
  amazonsales_treatment_after  <- log(amazonsales_treatment_after + 1)
  # t test
  t_test.amazonsales_treatment <- t.test(amazonsales_treatment_after, amazonsales_treatment_before)
  amazonsales_treatment_mean_diff_se <- t_test.amazonsales_treatment$stderr
  t_test.amazonsales_treatment$p.value
  amazonsales_treatment_after_mean <- t_test.amazonsales_treatment$estimate[["mean of x"]]
  amazonsales_treatment_before_mean <- t_test.amazonsales_treatment$estimate[["mean of y"]]
  amazonsales_treatment_mean_diff <- t_test.amazonsales_treatment$estimate[["mean of x"]] - t_test.amazonsales_treatment$estimate[["mean of y"]]

  # Amazon Sales DID
  amazonsales_did <- amazonsales_treatment_mean_diff - amazonsales_control_mean_diff

  # Amazon PagesPerDollar
  # for control
  amazonppd_control_before <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalPages / table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMonthlySales
  amazonppd_control_after <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalPages / table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_
```

```r
raw$AfterStoreClosing == 1),]$TotalMonthlySales

amazonppd_control_before <- log(amazonppd_control_before + 1)
amazonppd_control_after  <- log(amazonppd_control_after + 1)
# t test
t_test.amazonppd_control <- t.test(amazonppd_control_after, amazonppd_control_before)
amazonppd_control_mean_diff_se <- t_test.amazonppd_control$stderr
t_test.amazonppd_control$p.value
amazonppd_control_after_mean <- t_test.amazonppd_control$estimate[["mean of x"]]
amazonppd_control_before_mean <- t_test.amazonppd_control$estimate[["mean of y"]]
amazonppd_control_mean_diff <- t_test.amazonppd_control$estimate[["mean of x"]] - t_test.amazonppd_control$estimate[["mean of y"]]

# Amazon PagesPerDollar
# for treatment
amazonppd_treatment_before <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalPages / table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMonthlySales
amazonppd_treatment_after  <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalPages / table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalMonthlySales

amazonppd_treatment_before <- log(amazonppd_treatment_before + 1)
amazonppd_treatment_after  <- log(amazonppd_treatment_after + 1)
# t test
t_test.amazonppd_treatment <- t.test(amazonppd_treatment_after, amazonppd_treatment_before)
amazonppd_treatment_mean_diff_se <- t_test.amazonppd_treatment$stderr
t_test.amazonppd_treatment$p.value
amazonppd_treatment_after_mean <- t_test.amazonppd_treatment$estimate[["mean of x"]]
amazonppd_treatment_before_mean <-t_test.amazonppd_treatment$estimate[["mean of y"]]
amazonppd_treatment_mean_diff <- t_test.amazonppd_treatment$estimate[["mean of x"]] - t_test.amazonppd_treatment$estimate[["mean of y"]]

# Amazon PagesPerDollar DID
amazonppd_did <- amazonppd_treatment_mean_diff - amazonppd_control_mean_diff

# Amazon MinsPerDollar
# for control
amazonmpd_control_before <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMins / table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMonthlySales
amazonmpd_control_after  <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalMins / table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalMonthlySales

amazonmpd_control_before <- log(amazonmpd_control_before + 1)
amazonmpd_control_after  <- log(amazonmpd_control_after + 1)
# t test
t_test.amazonmpd_control <- t.test(amazonmpd_control_after, amazonmpd_control_before)
amazonmpd_control_mean_diff_se <- t_test.amazonmpd_control$stderr
t_test.amazonmpd_control$p.value
amazonmpd_control_after_mean <- t_test.amazonmpd_control$estimate[["mean of x"]]
amazonmpd_control_before_mean <- t_test.amazonmpd_control$estimate[["mean of y"]]
amazonmpd_control_mean_diff <- t_test.amazonmpd_control$estimate[["mean of x"]] - t_test.amazonmpd_control$estimate[["mean of y"]]

# Amazon MinsPerDollar
# for treatment
amazonmpd_treatment_before <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMins / table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 0),]$TotalMonthlySales
amazonmpd_treatment_after  <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalMins / table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$AfterStoreClosing == 1),]$TotalMonthlySales

amazonmpd_treatment_before <- log(amazonmpd_treatment_before + 1)
amazonmpd_treatment_after  <- log(amazonmpd_treatment_after + 1)
# t test
t_test.amazonmpd_treatment <- t.test(amazonmpd_treatment_after, amazonmpd_treatment_before)
amazonmpd_treatment_mean_diff_se <- t_test.amazonmpd_treatment$stderr
t_test.amazonmpd_treatment$p.value
amazonmpd_treatment_after_mean <- t_test.amazonmpd_treatment$estimate[["mean of x"]]
amazonmpd_treatment_before_mean <- t_test.amazonmpd_treatment$estimate[["mean of y"]]
amazonmpd_treatment_mean_diff <- t_test.amazonmpd_treatment$estimate[["mean of x"]] - t_test.amazonmpd_treatment$estimate[["mean of y"]]
```

```
  # Amazon MinsPerDollar DID
  amazonmpd_did <- amazonmpd_treatment_mean_diff - amazonmpd_control_mean_diff

  # construct table
  return(rbind(c(paste(print_name,"Sales"),"Control", amazonsales_control_after_mean, amazonsales_control_before_mean, amazonsales_control_mean_diff, amazonsales_control_mean_diff_se, amazonsales_did),
               c(paste(print_name,"Sales"),"Treatment", amazonsales_treatment_after_mean, amazonsales_treatment_before_mean, amazonsales_treatment_mean_diff, amazonsales_treatment_mean_diff_se, amazonsales_did),
               c(paste(print_name,"PagesPerDollar"),"Control", amazonppd_control_after_mean, amazonppd_control_before_mean, amazonppd_control_mean_diff, amazonppd_control_mean_diff_se, amazonppd_did),
               c(paste(print_name,"PagesPerDollar"),"Treatment", amazonppd_treatment_after_mean, amazonppd_treatment_before_mean, amazonppd_treatment_mean_diff, amazonppd_treatment_mean_diff_se, amazonppd_did),
               c(paste(print_name,"MinsPerDollar"),"Control", amazonmpd_control_after_mean, amazonmpd_control_before_mean, amazonmpd_control_mean_diff, amazonmpd_control_mean_diff_se, amazonmpd_did),
               c(paste(print_name,"MinsPerDollar"),"Treatment", amazonmpd_treatment_after_mean, amazonmpd_treatment_before_mean, amazonmpd_treatment_mean_diff, amazonmpd_treatment_mean_diff_se, amazonmpd_did))
  )
}

# generate table
amazon_table3 <- table3_gen(table3_0m_aggregate, "amazon.com", "Amazon")
bestbuy_table3 <- table3_gen(table3_0m_aggregate, "bestbuy.com", "bestbuy.com")

#
stargazer(rbind(amazon_table3, bestbuy_table3), align=TRUE, summary = FALSE, rownames = FALSE, title="Summary Statistics of Top Five Vendors by Sales Volume")
```

Code 4: Table 3 Generation

2.4 Table 4

In order to examine whether a competing online retailer benefits from the presence of a local showroom, we run the following regressions for Amazon.com and BestBuy.com:

$$\begin{aligned}
&\log\left(\texttt{TotalMonthlySales}+1\right)_{i,t}\\
&=\mu_i+\tau_t\\
&+\beta_1\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\\
&+\beta_2\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\times\texttt{BBStorePresent}_i\\
&+\epsilon_{i,t}
\end{aligned}\tag{1}$$

Table 4: Results of the Sales Effect (All Product Categories)

	log(TotalMonthlySales + 1)			
	Amazon-0 Mile	Amazon-5 Miles	BestBuy-0 Mile	BestBuy-5 Miles
	(1)	(2)	(3)	(4)
β_1	0.014	−0.005	−0.002	−0.002
	(0.015)	(0.008)	(0.033)	(0.008)
β_2	−0.033	0.003	0.009	0.002
	(0.022)	(0.010)	(0.036)	(0.010)
Observations	68,472	75,096	14,664	16,848
R^2	0.00003	0.00001	0.00002	0.00000
Adjusted R^2	-0.044	-0.044	-0.045	-0.045
F Statistic	1.091 (df = 2; 65594)	0.278 (df = 2; 71942)	0.154 (df = 2; 14028)	0.035 (df = 2; 16121)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 4 are listed below.

```
# Table 4 Data
data_0m_t4 <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1 GROUP BY Zip_Code, MonthYear, domain_name")
data_5m_t4 <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data2 GROUP BY Zip_Code, MonthYear, domain_name")
# manually construct DID and THREEINTERACTION
data_0m_t4$DID <- data_0m_t4$CCStorePresent * data_0m_t4$AfterStoreClosing
data_0m_t4$THREEINTER <- data_0m_t4$CCStorePresent * data_0m_t4$AfterStoreClosing * data_0m_t4$BBStorePresent
data_5m_t4$DID <- data_5m_t4$CCStorePresent * data_5m_t4$AfterStoreClosing
data_5m_t4$THREEINTER <- data_5m_t4$CCStorePresent * data_5m_t4$AfterStoreClosing * data_5m_t4$BBStorePresent
# Table 4
ama.t4.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t4[data_0m_t4$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t4.5mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t4[data_5m_t4$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t4.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t4[data_0m_t4$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t4.5mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t4[data_5m_t4$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 5: Table 4 Generation

2.5 Table 5

To measure the impact of the exit of local showrooms on consumer online search intensity and the moderating effect of Best Buy Stores as an alternative local showroom, we run the following regressions:

$$
\begin{aligned}
&\log\left(\texttt{PagesPerDollar}+1,\ \texttt{MinsPerDollar}+1\right)_{i,t}\\
&=\mu_i+\tau_t\\
&+\beta_1\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\\
&+\beta_2\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\times\texttt{BBStorePresent}_i\\
&+\epsilon_{i,t}
\end{aligned}
\tag{2}
$$

Table 5: Results of the Search Effect (All Product Categories)

	log(PagesPerDollar + 1)				log(MinsPerDollar + 1)			
	Amazon-0 Mile	Amazon-5 Miles	BestBuy-0 Mile	BestBuy-5 Miles	Amazon-0 Mile	Amazon-5 Miles	BestBuy-0 Mile	BestBuy-5 Miles
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
β_1	0.003	-0.019^{***}	0.001	0.002	0.004	-0.021^{***}	0.001	0.003
	(0.012)	(0.007)	(0.016)	(0.004)	(0.012)	(0.007)	(0.013)	(0.003)
β_2	-0.068^{***}	0.018^{**}	0.003	-0.001	-0.057^{***}	0.022^{***}	0.0004	-0.002
	(0.018)	(0.009)	(0.018)	(0.005)	(0.017)	(0.008)	(0.014)	(0.004)
Observations	68,472	75,096	14,664	16,848	68,472	75,096	14,664	16,848
R^2	0.0004	0.0001	0.00003	0.00004	0.0003	0.0001	0.00001	0.0001
Adjusted R^2	-0.043	-0.044	-0.045	-0.045	-0.044	-0.044	-0.045	-0.045
F Statistic	12.530^{***} (df = 2; 65594)	3.985^{**} (df = 2; 71942)	0.202 (df = 2; 14028)	0.337 (df = 2; 16121)	8.867^{***} (df = 2; 65594)	5.187^{***} (df = 2; 71942)	0.046 (df = 2; 14028)	0.451 (df = 2; 16121)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 5 are listed below.

```r
# Table 5 Data
data_0m_t5 <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1 GROUP BY Zip_Code, MonthYear, domain_name")
data_5m_t5 <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data2 GROUP BY Zip_Code, MonthYear, domain_name")
# manually construct DID and THREEINTERACTION
data_0m_t5$DID <- data_0m_t5$CCStorePresent * data_0m_t5$AfterStoreClosing
data_0m_t5$THREEINTER <- data_0m_t5$CCStorePresent * data_0m_t5$AfterStoreClosing * data_0m_t5$BBStorePresent
data_5m_t5$DID <- data_5m_t5$CCStorePresent * data_5m_t5$AfterStoreClosing
data_5m_t5$THREEINTER <- data_5m_t5$CCStorePresent * data_5m_t5$AfterStoreClosing * data_5m_t5$BBStorePresent
# Table 5
# For PagesPerDollar
ama.t5.pagesperdollar.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t5.pagesperdollar.5mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t5.pagesperdollar.0mile  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t5.pagesperdollar.5mile  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
# For MinsPerDollar
ama.t5.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t5.minsperdollar.5mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t5.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t5.minsperdollar.5mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 6: Table 5 Generation

2.6 Table 6

We then test whether the show-rooming effect upon online sales is stronger for experience goods, by grouping products into physical experience goods and search goods. Table 6 presents the results for sales model on physical experience products and search products.

Table 6: Results of the Sales Effect: Experience and Search Products

	log(TotalMonthlySales + 1)							
	Amazon-0 Mile-Exp	Amazon-5 Miles-Exp	Amazon-0 Mile-Search	Amazon-5 Miles-Search	BestBuy-0 Mile-Exp	BestBuy-5 Miles-Exp	BestBuy-0 Mile-Exp	BestBuy-5 Miles-Search
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
β_1	0.005	−0.007	0.005	−0.008	−0.011	−0.009	−0.001	−0.010
	(0.017)	(0.010)	(0.013)	(0.006)	(0.009)	(0.007)	(0.023)	(0.008)
β_2	−0.043*	0.009	−0.002	0.009		0.013	0.000	0.009
	(0.024)	(0.012)	(0.018)	(0.008)		(0.008)	(0.028)	(0.010)
Observations	32,112	35,568	52,392	57,648	10,224	11,712	5,664	6,600
R^2	0.0002	0.00002	0.00000	0.00003	0.0001	0.0002	0.00000	0.0002
Adjusted R^2	-0.044	-0.044	-0.044	-0.044	-0.046	-0.045	-0.048	-0.047
F Statistic	2.775* (df = 2; 30749)	0.318 (df = 2; 34061)	0.101 (df = 2; 50184)	0.774 (df = 2; 55221)	1.377 (df = 1; 9774)	1.297 (df = 2; 11199)	0.004 (df = 2; 5403)	0.746 (df = 2; 6300)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 6 are listed below.

```r
# Table 6 Data
data_0m_t6_exp <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1_exp GROUP BY Zip_Code, MonthYear, domain_name")
data_0m_t6_search <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1_search GROUP BY Zip_Code, MonthYear, domain_name")
data_5m_t6_exp <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data2_exp GROUP BY Zip_Code, MonthYear, domain_name")
data_5m_t6_search <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data2_search GROUP BY Zip_Code, MonthYear, domain_name")
# manually construct DID and THREEINTERACTION
data_0m_t6_exp$DID <- data_0m_t6_exp$CCStorePresent * data_0m_t6_exp$AfterStoreClosing
data_0m_t6_exp$THREEINTER <- data_0m_t6_exp$CCStorePresent * data_0m_t6_exp$AfterStoreClosing * data_0m_t6_exp$BBStorePresent
data_0m_t6_search$DID <- data_0m_t6_search$CCStorePresent * data_0m_t6_search$AfterStoreClosing
data_0m_t6_search$THREEINTER <- data_0m_t6_search$CCStorePresent * data_0m_t6_search$AfterStoreClosing * data_0m_t6_search$BBStorePresent
data_5m_t6_exp$DID <- data_5m_t6_exp$CCStorePresent * data_5m_t6_exp$AfterStoreClosing
data_5m_t6_exp$THREEINTER <- data_5m_t6_exp$CCStorePresent * data_5m_t6_exp$AfterStoreClosing * data_5m_t6_exp$BBStorePresent
data_5m_t6_search$DID <- data_5m_t6_search$CCStorePresent * data_5m_t6_search$AfterStoreClosing
data_5m_t6_search$THREEINTER <- data_5m_t6_search$CCStorePresent * data_5m_t6_search$AfterStoreClosing * data_5m_t6_search$BBStorePresent
# Table 6
# AmazonTotalMonthlySales & BBTotalMonthlySale vs Experience and Search Product
ama.t6.0mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_exp[data_0m_t6_exp$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t6.5mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_exp[data_5m_t6_exp$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t6.0mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_search[data_0m_t6_search$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t6.5mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_search[data_5m_t6_search$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t6.0mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_exp[data_0m_t6_exp$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t6.5mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_exp[data_5m_t6_exp$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t6.0mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_search[data_0m_t6_search$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t6.5mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_search[data_5m_t6_search$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 7: Table 6 Generation

2.7 Table 7

We then test whether the show-rooming effect upon online search behaviors is stronger for experience goods, by grouping products into physical experience goods and search goods. Table 7 and 8 presents the results for sales model on physical experience products and search products separately.

Table 7: Results of the Online Search Effect: Experience Products

	log(PagesPerDollar + 1)				log(MinsPerDollar + 1)			
	Amazon-0 Mile	Amazon-5 Miles	BestBuy-0 Mile	BestBuy-5 Miles	Amazon-0 Mile	Amazon-5 Miles	BestBuy-0 Mile	BestBuy-5 Miles
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
β_1	0.007	-0.037^{***}	0.006^{**}	0.001	0.006	-0.039^{***}	0.003	0.001
	(0.015)	(0.008)	(0.002)	(0.002)	(0.015)	(0.008)	(0.002)	(0.001)
β_2	-0.077^{***}	0.030^{***}		-0.0001	-0.067^{***}	0.034^{***}		-0.001
	(0.020)	(0.010)		(0.002)	(0.020)	(0.010)		(0.002)
Observations	32,112	35,568	10,224	11,712	32,112	35,568	10,224	11,712
R^2	0.001	0.001	0.001	0.00003	0.001	0.001	0.0003	0.0001
Adjusted R^2	-0.043	-0.044	-0.045	-0.046	-0.044	-0.044	-0.046	-0.046
F Statistic	12.857^{***} (df = 2; 30749)	10.009^{***} (df = 2; 34061)	5.763^{**} (df = 1; 9774)	0.143 (df = 2; 11199)	10.349^{***} (df = 2; 30749)	11.626^{***} (df = 2; 34061)	2.508 (df = 1; 9774)	0.438 (df = 2; 11199)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 7 are listed below.

```
# Table 7 & 8 Data
data_0m_t7_exp    <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1_exp GROUP BY Zip_Code, MonthYear, domain_name")
data_0m_t8_search <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1_search GROUP BY Zip_Code, MonthYear, domain_name")
data_5m_t7_exp    <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data2_exp GROUP BY Zip_Code, MonthYear, domain_name")
data_5m_t8_search <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data2_search GROUP BY Zip_Code, MonthYear, domain_name")
# manually construct DID and THREEINTERACTION
data_0m_t7_exp$DID <- data_0m_t7_exp$CCStorePresent * data_0m_t7_exp$AfterStoreClosing
data_0m_t7_exp$THREEINTER <- data_0m_t7_exp$CCStorePresent * data_0m_t7_exp$AfterStoreClosing * data_0m_t7_exp$BBStorePresent
data_0m_t8_search$DID <- data_0m_t8_search$CCStorePresent * data_0m_t8_search$AfterStoreClosing
data_0m_t8_search$THREEINTER <- data_0m_t8_search$CCStorePresent * data_0m_t8_search$AfterStoreClosing * data_0m_t8_search$BBStorePresent
data_5m_t7_exp$DID <- data_5m_t7_exp$CCStorePresent * data_5m_t7_exp$AfterStoreClosing
data_5m_t7_exp$THREEINTER <- data_5m_t7_exp$CCStorePresent * data_5m_t7_exp$AfterStoreClosing * data_5m_t7_exp$BBStorePresent
data_5m_t8_search$DID <- data_5m_t8_search$CCStorePresent * data_5m_t8_search$AfterStoreClosing
data_5m_t8_search$THREEINTER <- data_5m_t8_search$CCStorePresent * data_5m_t8_search$AfterStoreClosing * data_5m_t8_search$BBStorePresent
# Table 7
ama.t7.pagesperdollar.0mile.exp <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t7.pagesperdollar.5mile.exp <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t7_exp[data_5m_t7_exp$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t7.pagesperdollar.0mile.exp  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t7.pagesperdollar.5mile.exp  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t7_exp[data_5m_t7_exp$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t7.minsperdollar.0mile.exp  <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t7.minsperdollar.5mile.exp  <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t7_exp[data_5m_t7_exp$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t7.minsperdollar.0mile.exp   <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t7.minsperdollar.5mile.exp   <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t7_exp[data_5m_t7_exp$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 8: Table 7 Generation

2.8 Table 8

Table 8: Results of the Online Search Effect: Search Products

	log(PagesPerDollar + 1)				log(MinsPerDollar + 1)			
	Amazon-0 Mile	Amazon-5 Miles	BestBuy-0 Mile	BestBuy-5 Miles	Amazon-0 Mile	Amazon-5 Miles	BestBuy-0 Mile	BestBuy-5 Miles
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
β_1	0.001	0.006	0.001	0.009*	0.003	0.004	0.0001	0.009*
	(0.012)	(0.006)	(0.014)	(0.005)	(0.012)	(0.006)	(0.012)	(0.005)
β_2	−0.019	−0.002	−0.000	−0.007	−0.019	0.001	−0.000	−0.008
	(0.017)	(0.008)	(0.017)	(0.006)	(0.017)	(0.008)	(0.015)	(0.006)
Observations	52,392	57,648	5,664	6,600	52,392	57,648	5,664	6,600
R^2	0.00005	0.00002	0.00000	0.001	0.00004	0.00003	0.00000	0.001
Adjusted R^2	-0.044	-0.044	-0.048	-0.047	-0.044	-0.044	-0.048	-0.047
F Statistic	1.138 (df = 2; 50184)	0.553 (df = 2; 55221)	0.011 (df = 2; 5403)	1.590 (df = 2; 6300)	0.935 (df = 2; 50184)	0.696 (df = 2; 55221)	0.0001 (df = 2; 5403)	1.927 (df = 2; 6300)
Note:								*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 8 are listed below.

```
# Table 8
ama.t8.pagesperdollar.0mile.search <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t8_search[data_0m_t8_search$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t8.pagesperdollar.5mile.search <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t8_search[data_5m_t8_search$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t8.pagesperdollar.0mile.search  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t8_search[data_0m_t8_search$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t8.pagesperdollar.5mile.search  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t8_search[data_5m_t8_search$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t8.minsperdollar.0mile.search  <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t8_search[data_0m_t8_search$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t8.minsperdollar.5mile.search  <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t8_search[data_5m_t8_search$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t8.minsperdollar.0mile.search   <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t8_search[data_0m_t8_search$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t8.minsperdollar.5mile.search   <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t8_search[data_5m_t8_search$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 9: Table 8 Generation

2.9 Table 9

To capture the expected change in the odds ratio of the impact of Circuit City store closures and the moderating effect of Best Buy stores as an alternative local showroom, we run the following regressions:

$$\begin{aligned}
&\text{Logit}\left(\texttt{ReferringDomainIsSearchEngine}, \texttt{NoReferringDomain}\right)_{i,t} \\
&= \mu_i + \tau_t \\
&+ \beta_1\ \texttt{CCStorePresent}_i \times \texttt{AfterStoreClosing}_t \\
&+ \beta_2\ \texttt{CCStorePresent}_i \times \texttt{AfterStoreClosing}_t \times \texttt{BBStorePresent}_i \\
&+ \epsilon_{i,t}
\end{aligned} \tag{3}$$

Table 9 presents the effect of store closure on referring domain.

Table 9: Results of Logistic Regression for Referring Domain

	ReferringDomainIsSearchEngine		NoReferringDomain	
	Amazon-0 Mile	BestBuy-0 Mile	Amazon-0 Mile	BestBuy-0 Mile
	(1)	(2)	(3)	(4)
β_1	-0.817^{*}	-15.12^{***}	0.325	-0.223
	(0.337)	(0.611)	(0.346)	(1.259)
β_2	0.697	14.43^{***}	-0.415	0.916
	(0.564)	(0.944)	(0.544)	(1.615)
Observations	10,791	1,225	10,791	1,225

Note: *p<0.05; **p<0.01; ***p<0.001

Stata codes for generating Table 9 are listed below.

```
* build variables
gen DID =  CCStorePresent *  AfterStoreClosing
gen THREEINTER = DID *  BBStorePresent
egen Code_Time = group(Zip_Code MonthYear)

* Amazon - ReferringDomainIsSearchEngine & NoReferringDomain
eststo: logit ReferringDomainIsSearchEngine DID THREEINTER if domain_name == "amazon.com", vce(cluster Code_Time) noconstant
eststo: logit NoReferringDomain DID THREEINTER if domain_name == "amazon.com", vce(cluster Code_Time) noconstant
* BestBuy - ReferringDomainIsSearchEngine & NoReferringDomain
eststo: logit ReferringDomainIsSearchEngine DID THREEINTER if domain_name == "bestbuy.com", vce(cluster MonthYear) noconstant
eststo: logit NoReferringDomain DID THREEINTER if domain_name == "bestbuy.com", vce(cluster MonthYear) noconstant
```

Code 10: Table 9 Generation

2.10 Table 10

By applying more traditional online search measures, we perform the same DID analysis for Amazon and bestbuy.com, to further investigate if the increase in search intensity manifests itself independent of sales amount.

Table 10: Results of the Online Sales and Search Effect (All Product Categories)

	log(SalesPerTransaction + 1)		log(PagesPerTransaction + 1)		log(MinsPerTransaction + 1)	
	Amazon-0 Mile	BestBuy-0 Mile	Amazon-0 Mile	BestBuy-0 Mile	Amazon-0 Mile	BestBuy-0 Mile
	(1)	(2)	(3)	(4)	(5)	(6)
β_1	0.012	−0.001	0.004	0.0002	0.006	0.0002
	(0.013)	(0.032)	(0.009)	(0.017)	(0.011)	(0.020)
β_2	−0.018	0.010	-0.021^{*}	0.005	−0.021	−0.003
	(0.019)	(0.034)	(0.013)	(0.018)	(0.016)	(0.021)
Observations	68,472	14,664	68,472	14,664	68,472	14,664
R^2	0.00002	0.00003	0.0001	0.00004	0.00003	0.00001
Adjusted R^2	-0.044	-0.045	-0.044	-0.045	-0.044	-0.045
F Statistic	0.539 (df = 2; 65594)	0.213 (df = 2; 14028)	1.867 (df = 2; 65594)	0.304 (df = 2; 14028)	0.939 (df = 2; 65594)	0.066 (df = 2; 14028)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 10 are listed below.

```r
# Table 10 Data
data_0m_t10 <- sqldf("SELECT Zip_Code, MonthYear, domain_name, AVG(pages_viewed) AS PagesPerTransaction, AVG(duration) AS MinsPerTransaction, AVG(prod_totprice) AS SalesPerTransaction, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1 GROUP BY Zip_Code, MonthYear, domain_name")

data_0m_t10$DID <- data_0m_t10$CCStorePresent * data_0m_t10$AfterStoreClosing
data_0m_t10$THREEINTER <- data_0m_t10$CCStorePresent * data_0m_t10$AfterStoreClosing * data_0m_t10$BBStorePresent
# Table 10
# SalesPerTransaction; PagesPerTransaction; MinsPerTransaction; for Ama & BB
ama.t10.0mile.SalesPerTransaction <- plm(log(SalesPerTransaction + 1) ~ DID + THREEINTER, data = data_0m_t10[data_0m_t10$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t10.0mile.SalesPerTransaction  <- plm(log(SalesPerTransaction + 1) ~ DID + THREEINTER, data = data_0m_t10[data_0m_t10$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t10.0mile.PagesPerTransaction <- plm(log(PagesPerTransaction + 1) ~ DID + THREEINTER, data = data_0m_t10[data_0m_t10$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t10.0mile.PagesPerTransaction  <- plm(log(PagesPerTransaction + 1) ~ DID + THREEINTER, data = data_0m_t10[data_0m_t10$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t10.0mile.MinsPerTransaction <- plm(log(MinsPerTransaction + 1) ~ DID + THREEINTER, data = data_0m_t10[data_0m_t10$domain_name == "amazon.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t10.0mile.MinsPerTransaction  <- plm(log(MinsPerTransaction + 1) ~ DID + THREEINTER, data = data_0m_t10[data_0m_t10$domain_name == "bestbuy.com",], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 11: Table 10 Generation

2.11 Table 11

To further investigate if the increase in search intensity has a causal link to the Circuit City store closures and not due to other endogenous reasons, we adopt coarsened exact matching algorithm to match each zip code from the treatment group with an equivalent zip code from the control group, using zip code level demographics (average household age, average income and average household size). The matching results left us with 56 zip codes in each group. Using the data from the combined 112 zip codes, we ran the models for sales and search.

Table 11: Results of the Online Sales and Search Effect After Matching Zip Codes: TotalMonthlySales, PagesPerDollar, and MinsPerDollar (All Product Categories)

	log(TotalMonthlySales + 1)		log(PagesPerDollar + 1)		log(MinsPerDollar + 1)	
	Amazon-0 Mile	BesyBuy-0 Mile	Amazon-0 Mile	BesyBuy-0 Mile	Amazon-0 Mile	BesyBuy-0 Mile
	(1)	(2)	(3)	(4)	(5)	(6)
β_1	0.019	−0.0002	0.006	−0.001	0.003	−0.0002
	(0.019)	(0.002)	(0.012)	(0.003)	(0.011)	(0.002)
β_2	−0.026		−0.023		−0.024*	
	(0.024)		(0.016)		(0.013)	
Observations	1,776	384	1,776	384	1,776	384
R^2	0.001	0.00002	0.001	0.0001	0.002	0.00003
Adjusted R^2	-0.058	-0.113	-0.057	-0.113	-0.056	-0.113
F Statistic	0.740 (df = 2; 1677)	0.008 (df = 1; 344)	1.183 (df = 2; 1677)	0.030 (df = 1; 344)	1.931 (df = 2; 1677)	0.012 (df = 1; 344)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 11 are listed below.

```
library(cem)
#matching based on zipcode demographics (cross-sectional)
data_0m_t11 <- sqldf("SELECT Zip_Code, SUM(prod_totprice) AS TotalMonthlySales,
                     AVG(CCStorePresent) AS CCStorePresent,
                     AVG(household_size) AS HoHSize,
                     AVG(hoh_oldest_age) AS HoHAge,
                     AVG(household_income) AS HoHIncome,
                     AVG(children) AS HoHChildren,
                     AVG(connection_speed) AS HoHSpeed
                     FROM concat_data1 GROUP BY Zip_Code")

#check imblance within data set
vars <- c("HoHSize", "HoHAge", "HoHIncome", "HoHChildren", "HoHSpeed")
imbalance(group=data_0m_t11$CCStorePresent, data = data_0m_t11[vars])

# Default is not 1-1 matching in CEM. Use k2k = "True" to enforce 1 to 1 matching.
todrop <- c("TotalMonthlySales")
todrop2 <- c("TotalMonthlySales", "Zip_Code")
# mat <- cem(treatment = "CCStorePresent", data = data_0m_t11, drop = todrop, k2k ="True")

mat <- cem(treatment = "CCStorePresent",
           data = data_0m_t11,
           drop = todrop2,
           k2k = TRUE,
           method = "euclidean")
mat
```

```r

# We got 110 zipcodes in total. We checked 2 dataframe from CEM results, "w" and "matched", and both have 110 values.
# Fortunately, they are the same. In the future, just use data from "matched". Note that this is only ID of row value of Zipcode

# assign ID of row value of zipcode from "matched"
zipcheck <- c()

for (i in 1:length(mat$matched)){
  if (mat$matched[i] == "TRUE") zipcheck <-c(zipcheck,i)
}

data.frame(zipcheck)

# assign ID of row value of zipcode from "w"
zipcheck1 <- c()

for (i in 1:length(mat$w)){
  if (mat$w[i] == 1) zipcheck1 <-c(zipcheck1,i)
}

data.frame(zipcheck1)

# Test both dataframe, and they are same.
all.equal(zipcheck,zipcheck1)

# add specific Zipcode by mapping from ID of row of matched zipcode
ziplist <- c()
for (i in 1:length(data_0m_t11$Zip_Code)){
  if ( i %in% zipcheck) ziplist <-c(ziplist,data_0m_t11$Zip_Code[i])
}

data.frame(ziplist)

# assign matched zipcode to dataset
concat_data1$Zipmatch <- ifelse(concat_data1$Zip_Code %in% ziplist, 1, 0)
data_0m_t11 <- sqldf("SELECT Zip_Code, Zipmatch, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1 GROUP BY Zip_Code, MonthYear, domain_name")
data_0m_t11$DID <- data_0m_t11$CCStorePresent * data_0m_t11$AfterStoreClosing
data_0m_t11$THREEINTER <- data_0m_t11$DID * data_0m_t11$BBStorePresent

# result for Amazon regarding TotalMonthlySales, PagesPerDollar, MinsPerDollar
ama.t11.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t11[(data_0m_t11$domain_name == "amazon.com") & (data_0m_t11$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t11.pagesperdollar.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t11[(data_0m_t11$domain_name == "amazon.com") & (data_0m_t11$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t11.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t11[(data_0m_t11$domain_name == "amazon.com") & (data_0m_t11$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
# result for Bestbuy regarding TotalMonthlySales, PagesPerDollar, MinsPerDollar
bb.t11.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t11[(data_0m_t11$domain_name == "bestbuy.com") & (data_0m_t11$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t11.pagesperdollar.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t11[(data_0m_t11$domain_name == "bestbuy.com") & (data_0m_t11$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t11.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t11[(data_0m_t11$domain_name == "bestbuy.com") & (data_0m_t11$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 12: Table 11 Generation

2.12 Table 12

To examine the possible heterogeneity within the geographic zip code area which may be unaccounted for, we add location specific demographics in the regression equations as interaction terms with DID term.

Table 12: Results of the Online Sales and Search Effect with Zip Code Demographics as Interactions and Time Fixed Effects (All Product Categories)

	log(TotalMonthlySales + 1)		log(PagesPerDollar + 1)		log(MinsPerDollar + 1)	
	Amazon-0 Mile	BestBuy-0 Mile	Amazon-0 Mile	BestBuy-0 Mile	Amazon-0 Mile	BestBuy-0 Mile
	(1)	(2)	(3)	(4)	(5)	(6)
β_1	−0.00001	−0.00001	0.0001	0.00001	0.0001	0.00000
	(0.0001)	(0.0002)	(0.0001)	(0.0001)	(0.0001)	(0.0001)
β_2	−0.00001	−0.0001	−0.0002*	0.00002	−0.0002	0.00001
	(0.0002)	(0.0002)	(0.0001)	(0.0001)	(0.0001)	(0.0001)
Observations	68,472	14,664	68,472	14,664	68,472	14,664
R^2	0.00000	0.00004	0.00005	0.00002	0.00003	0.00001
Adjusted R^2	-0.044	-0.045	-0.044	-0.045	-0.044	-0.045
F Statistic	0.019 (df = 2; 65594)	0.255 (df = 2; 14028)	1.478 (df = 2; 65594)	0.114 (df = 2; 14028)	1.131 (df = 2; 65594)	0.053 (df = 2; 14028)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 12 are listed below.

```
# Table 12 Data
data_0m_t12 <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, SUM(pages_viewed) / SUM(prod_totprice) AS
      PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(household_size) AS HoHSize, AVG(hoh_oldest_age) AS HoHAge, AVG(
      household_income) AS HoHIncome, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS
      AfterStoreClosing FROM concat_data1 GROUP BY Zip_Code, MonthYear, domain_name")
data_0m_t12$DID <- data_0m_t12$CCStorePresent * data_0m_t12$AfterStoreClosing * data_0m_t12$HoHSize * data_0m_t12$HoHAge * data_0m_t12$HoHIncome
data_0m_t12$THREEINTER <- data_0m_t12$CCStorePresent * data_0m_t12$AfterStoreClosing * data_0m_t12$BBStorePresent
# Table 12
ama.t12.0m.PagesPerDollar <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t12[data_0m_t12$domain_name == "amazon.com",], index =
      c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t12.0m.MinsPerDollar <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t12[data_0m_t12$domain_name == "amazon.com",], index = c(
      "Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.t12.0m.TotalMonthlySales <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t12[data_0m_t12$domain_name == "amazon.com",],
      index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t12.0m.PagesPerDollar <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t12[data_0m_t12$domain_name == "bestbuy.com",], index =
      c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t12.0m.MinsPerDollar <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t12[data_0m_t12$domain_name == "bestbuy.com",], index = c(
      "Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.t12.0m.TotalMonthlySales <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t12[data_0m_t12$domain_name == "bestbuy.com",],
      index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
```

Code 13: Table 12 Generation

2.13 Table 13

In order to address the serial correlation issue, the first solution is to ignore the time series and the results is in Table 13.

Table 13: Results of the Online Sales and Search Effect After Matching Zip Codes: TotalMonthlySales, PagesPerDollar, and MinsPerDollar (All Product Categories)

	log(TotalMonthlySales + 1)		log(PagesPerDollar + 1)		log(MinsPerDollar + 1)	
	Amazon-0 Mile	BesyBuy-0 Mile	Amazon-0 Mile	BesyBuy-0 Mile	Amazon-0 Mile	BesyBuy-0 Mile
	(1)	(2)	(3)	(4)	(5)	(6)
β_1	0.023	−0.001	0.007	−0.003	0.009	−0.001
	(0.015)	(0.003)	(0.010)	(0.006)	(0.008)	(0.004)
β_2	−0.026		−0.023		−0.024*	
	(0.024)		(0.015)		(0.013)	
Observations	1,776	208	1,776	208	1,776	208
R^2	0.001	0.0003	0.001	0.001	0.002	0.0004
Adjusted R^2	-0.043	-0.083	-0.043	-0.083	-0.042	-0.083
F Statistic	1.169 (df = 2; 1700)	0.052 (df = 1; 191)	1.182 (df = 2; 1700)	0.197 (df = 1; 191)	1.663 (df = 2; 1700)	0.078 (df = 1; 191)
Note:						*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 13 are listed below.

```
library(cem)
#matching based on zipcode demographics (cross-sectional)
data_0m_t11 <- sqldf("SELECT Zip_Code, SUM(prod_totprice) AS TotalMonthlySales,
                     AVG(CCStorePresent) AS CCStorePresent,
                     AVG(household_size) AS HoHSize,
                     AVG(hoh_oldest_age) AS HoHAge,
                     AVG(household_income) AS HoHIncome,
                     AVG(children) AS HoHChildren,
                     AVG(connection_speed) AS HoHSpeed
                     FROM concat_data1 GROUP BY Zip_Code")
# CEM
todrop2 <- c("TotalMonthlySales", "Zip_Code")
mat <- cem(treatment = "CCStorePresent",
           data = data_0m_t11,
           drop = todrop2,
           k2k = TRUE,
           method = "euclidean")
mat

# Check Matching
zipcheck <- c()

for (i in 1:length(mat$matched)){
  if (mat$matched[i] == "TRUE") zipcheck <-c(zipcheck,i)
}

data.frame(zipcheck)

# assign ID of row value of zipcode from "w"
zipcheck1 <- c()

for (i in 1:length(mat$w)){
  if (mat$w[i] == 1) zipcheck1 <-c(zipcheck1,i)
}
```

```
data.frame(zipcheck1)

# Test both dataframe, and they are same.
all.equal(zipcheck,zipcheck1)

# add specific Zipcode by mapping from ID of row of matched zipcode
ziplist <- c()
for (i in 1:length(data_0m_t11$Zip_Code)){
  if ( i %in% zipcheck) ziplist <-c(ziplist,data_0m_t11$Zip_Code[i])
}

data.frame(ziplist)

# Assign matched zipcode to dataset
concat_data1$Zipmatch <- ifelse(concat_data1$Zip_Code %in% ziplist, 1, 0)
data_0m_t13 <- sqldf("SELECT Zip_Code, Zipmatch, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, SUM(pages_viewed) / SUM(prod_totprice) AS PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1 GROUP BY Zip_Code, MonthYear, domain_name")
data_0m_t13$DID <- data_0m_t13$CCStorePresent * data_0m_t13$AfterStoreClosing
data_0m_t13$THREEINTER <- data_0m_t13$DID * data_0m_t13$BBStorePresent

# Table 13
ama.0m.t13.sales <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "amazon.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code"), model = "within")
ama.0m.t13.ppd   <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "amazon.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code"), model = "within")
ama.0m.t13.mpd   <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "amazon.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code"), model = "within")

bb.0m.t13.sales <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "bestbuy.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code"), model = "within")
bb.0m.t13.ppd   <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "bestbuy.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code"), model = "within")
bb.0m.t13.mpd   <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "bestbuy.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code"), model = "within")
```

Code 14: Table 13 Generation

2.14 Table 14

For the serial correlation issue, another solution is to to use a White-like estimator to calculate the variance-covariance matrix of the error term.The results is in Table 14.

Table 14: Results of the Online Sales and Search Effect with Arbitrary Variance-Covariance Matrix Corrections (All Product Categories)

	log(TotalMonthlySales + 1)		log(PagesPerDollar + 1)		log(MinsPerDollar + 1)	
	Amazon-0 Mile	BestBuy-0 Mile	Amazon-0 Mile	BestBuy-0 Mile	Amazon-0 Mile	BestBuy-0 Mile
	(1)	(2)	(3)	(4)	(5)	(6)
β_1	0.019	−0.001	0.006	−0.003	0.003	−0.001
	(0.019)	(0.003)	(0.006)	(0.006)	(0.011)	(0.004)
β_2	−0.026		-0.023^{***}		-0.024^{***}	
	(0.028)		(0.001)		(0.006)	
Observations	1,776	208	1,776	208	1,776	208
R^2	0.001	0.0002	0.001	0.001	0.002	0.0003
Adjusted R^2	-0.058	-0.156	-0.057	-0.156	-0.056	-0.156
F Statistic	0.740 (df = 2; 1677)	0.036 (df = 1; 179)	1.183 (df = 2; 1677)	0.136 (df = 1; 179)	1.931 (df = 2; 1677)	0.054 (df = 1; 179)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 14 are listed below.

```
# Table 14
library(lmtest)
library(sandwich)

# Create Baseline
ama.0m.t14.sale.base <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "amazon.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.0m.t14.ppd.base  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "amazon.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
ama.0m.t14.mpd.base  <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "amazon.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")

bb.0m.t14.sale.base <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "bestbuy.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.0m.t14.ppd.base  <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "bestbuy.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
bb.0m.t14.mpd.base  <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t13[(data_0m_t13$domain_name == "bestbuy.com") & (data_0m_t13$Zipmatch == 1),], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")

# Correlation
coeftest(ama.0m.t14.sale.base, vcovDC)
coeftest(ama.0m.t14.ppd.base, vcovDC)
coeftest(ama.0m.t14.mpd.base, vcovDC)
```

Code 15: Table 14 Generation

2.15 Table C1

Table 15: Change in Demographics after Circuit City Store Closure

Group	Before Store Closure			After Store Closure			First Difference of Mean (p-value)		
	Mean Age	Mean Income	Mean Education	Mean Age	Mean Income	Mean Education	Mean Age	Mean Income	Mean Education
Control	7.048	4.479	97.957	6.937	4.498	97.999	-0.111 (<0.0001)	0.019 (0.300)	0.042 (0.639)
Treated	7.68	4.971	98.632	6.645	4.739	96.843	-1.035 (<0.0001)	-0.232 (0.029)	-1.789 (0.004)

Codes for generating Table 15 are listed below.

```
temp <- read_sas(sales_allother_zipcode_path)
temp$Store_Close_Status <- 0
table_C1_0m_raw <-  rbind(temp, read_sas(sales_cc_0mile_path))
table_C1_5m_raw <-  rbind(temp, read_sas(sales_cc_5miles_path))

# Date Transform
table_C1_0m_raw$event_date <- as.Date(table_C1_0m_raw$event_date)
table_C1_5m_raw$event_date <- as.Date(table_C1_5m_raw$event_date)

# construct MonthYear - month of year
table_C1_0m_raw$MonthYear <- format(table_C1_0m_raw$event_date, "%Y-%m")
table_C1_5m_raw$MonthYear <- format(table_C1_5m_raw$event_date, "%Y-%m")

# Mark CC Closure

# CCStorePresent
# it is the same as Store_Close_Status
table_C1_0m_raw$CCStorePresent <- table_C1_0m_raw$Store_Close_Status
table_C1_5m_raw$CCStorePresent <- table_C1_5m_raw$Store_Close_Status

# AfterStoreClosing
table_C1_0m_raw$AfterStoreClosing <- ifelse(table_C1_0m_raw$MonthYear < "2008-11", 0, 1)
table_C1_5m_raw$AfterStoreClosing <- ifelse(table_C1_5m_raw$MonthYear < "2008-11", 0, 1)

# BBStorePresent
table_C1_0m_raw <- merge(table_C1_0m_raw, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)
table_C1_5m_raw <- merge(table_C1_5m_raw, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)

table_C1_0m_raw$BBStorePresent <- na.fill(table_C1_0m_raw$BB_Store_Status, 0)
table_C1_5m_raw$BBStorePresent <- na.fill(table_C1_5m_raw$BB_Store_Status, 0)

# t test
control_before_age    <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 0)&(table_C1_0m_raw$AfterStoreClosing==0),]$hoh_oldest_age
control_before_income <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 0)&(table_C1_0m_raw$AfterStoreClosing==0),]$household_income
control_before_edu    <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 0)&(table_C1_0m_raw$AfterStoreClosing==0),]$hoh_most_education

control_after_age    <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 0)&(table_C1_0m_raw$AfterStoreClosing==1),]$hoh_oldest_age
control_after_income <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 0)&(table_C1_0m_raw$AfterStoreClosing==1),]$household_income
control_after_edu    <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 0)&(table_C1_0m_raw$AfterStoreClosing==1),]$hoh_most_education

test.control.age <- t.test(control_before_age, control_after_age)
test.control.income <- t.test(control_before_income, control_after_income)
test.control.edu <- t.test(control_before_edu, control_after_edu)

treated_before_age    <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 1)&(table_C1_0m_raw$AfterStoreClosing==0),]$hoh_oldest_age
treated_before_income <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 1)&(table_C1_0m_raw$AfterStoreClosing==0),]$household_income
treated_before_edu    <- table_C1_0m_raw[(table_C1_0m_raw$CCStorePresent == 1)&(table_C1_0m_raw$AfterStoreClosing==0),]$hoh_most_education
```

```
treated_after_age    <- table_C1_Om_raw[(table_C1_Om_raw$CCStorePresent == 1)&(table_C1_Om_raw$AfterStoreClosing==1),]$hoh_oldest_age
treated_after_income <- table_C1_Om_raw[(table_C1_Om_raw$CCStorePresent == 1)&(table_C1_Om_raw$AfterStoreClosing==1),]$household_income
treated_after_edu    <- table_C1_Om_raw[(table_C1_Om_raw$CCStorePresent == 1)&(table_C1_Om_raw$AfterStoreClosing==1),]$hoh_most_education

test.treated.age <- t.test(treated_before_age, treated_after_age)
test.treated.income <- t.test(treated_before_income, treated_after_income)
test.treated.edu <- t.test(treated_before_edu, treated_after_edu)

# Construct Variables
test.control.age.before.mean    <- test.control.age$estimate[["mean of x"]]
test.control.income.before.mean <- test.control.income$estimate[["mean of x"]]
test.control.edu.before.mean    <- test.control.edu$estimate[["mean of x"]]

test.control.age.after.mean    <- test.control.age$estimate[["mean of y"]]
test.control.income.after.mean <- test.control.income$estimate[["mean of y"]]
test.control.edu.after.mean    <- test.control.edu$estimate[["mean of y"]]

test.control.age.diff    <- test.control.age.after.mean - test.control.age.before.mean
test.control.age.diff.se <- test.control.age$p.value
test.control.income.diff    <- test.control.income.after.mean - test.control.income.before.mean
test.control.income.diff.se <- test.control.income$p.value
test.control.edu.diff    <- test.control.edu.after.mean - test.control.edu.before.mean
test.control.edu.diff.se <- test.control.edu$p.value

test.treated.age.before.mean    <- test.treated.age$estimate[["mean of x"]]
test.treated.income.before.mean <- test.treated.income$estimate[["mean of x"]]
test.treated.edu.before.mean    <- test.treated.edu$estimate[["mean of x"]]

test.treated.age.after.mean    <- test.treated.age$estimate[["mean of y"]]
test.treated.income.after.mean <- test.treated.income$estimate[["mean of y"]]
test.treated.edu.after.mean    <- test.treated.edu$estimate[["mean of y"]]

test.treated.age.diff    <- test.treated.age.after.mean - test.treated.age.before.mean
test.treated.age.diff.se <- test.treated.age$p.value
test.treated.income.diff    <- test.treated.income.after.mean - test.treated.income.before.mean
test.treated.income.diff.se <- test.treated.income$p.value
test.treated.edu.diff    <- test.treated.edu.after.mean - test.treated.edu.before.mean
test.treated.edu.diff.se <- test.treated.edu$p.value

# Construct Table
tabc1 <- rbind(c("Control", test.control.age.before.mean, test.control.income.before.mean, test.control.edu.before.mean,
                   test.control.age.after.mean, test.control.income.after.mean, test.control.edu.after.mean,
                   test.control.age.diff, test.control.age.diff.se, test.control.income.diff, test.control.income.diff.se, test.control.edu.diff,
      test.control.edu.diff.se),
      c("Treated", test.treated.age.before.mean, test.treated.income.before.mean, test.treated.edu.before.mean,
                   test.treated.age.after.mean, test.treated.income.after.mean, test.treated.edu.after.mean,
                   test.treated.age.diff, test.treated.age.diff.se, test.treated.income.diff, test.treated.income.diff.se, test.treated.edu.diff,
      test.treated.edu.diff.se))
```

Code 16: Table C1 Generation

2.16 Table D1-D3

In this section, we further investigate the relationship between search and sales to understand the underlying conversion.

2.16.1 Table D1

In order to capture how much time a user would spend on a page on average before making a purchase, we define a new search intensity measure called minutes per page. And we run the following model to explore the relationship between search intensity and change in sales:

$$\begin{aligned} &\log(\texttt{Sales}+1)_{i,t} \\ &= \mu_i + \tau_t \\ &+ \beta_1\ \texttt{MinsPerPage}_{i,t} \\ &+ \beta_2\ \texttt{ExperienceGood}_{i,t} \\ &+ \beta_3\ \texttt{MinsPerPage}_{i,t} \times \texttt{ExperienceGood}_{i,t} \\ &+ \epsilon_{i,t} \end{aligned} \tag{4}$$

Table 16: Search Intensity Effects on Sales for Amazon

	(1) Log(Sales + 1)
β_1	2.376***
	(0.0435)
β_2	3.194***
	(0.0675)
β_3	-2.153***
	(0.0744)
Observations	10791

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Stata codes for generating Table 16 are listed below.

```
eststo: reg LogSales MinsPerPage ExperienceGood tD1INTER if domain_name == "amazon.com", vce(cluster Code_Time) noconstant

esttab using tableD1.tex, se label replace booktabs title(Search Intensity Effects on Sales for Amazon\label{tab:tabD1})

eststo clear
```

Code 17: Table D1 Generation

2.16.2 Table D2

Next, we explored the correlation between product characteristics and search intensity during a transaction for Amazon sales:

$$\begin{aligned}&\log\left(\texttt{PagesViewed}, \texttt{MinsSpent}+1\right)_{i,t}\\&=\mu_i+\tau_t\\&+\beta_1\ \texttt{ProductPrice}_{i,t}\\&+\beta_2\ \texttt{ExperienceGood}_{i,t}\\&+\epsilon_{i,t}\end{aligned} \tag{5}$$

Table 17: Product Characteristics Effects on Search Intensity for Amazon

	(1) Log(PagesViewed)	(2) Log(MinsSpent + 1)
β_1	0.00465***	0.00450***
	(0.000843)	(0.000785)
β_2	3.156***	2.923***
	(0.0581)	(0.0558)
Observations	10791	10791

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Stata codes for generating Table 17 are listed below.

```
eststo: reg LogPagesViewed prod_totprice ExperienceGood if domain_name == "amazon.com", vce(cluster Code_Time) noconstant

eststo: reg LogMinsSpent prod_totprice ExperienceGood if domain_name == "amazon.com", vce(cluster Code_Time) noconstant

esttab using tableD2.tex, se label replace booktabs title(Product Characteristics Effects on Search Intensity for Amazon\label{tab:tabD2})

eststo clear
```

Code 18: Table D2 Generation

2.16.3 Table D3

To study the search path used by experienced goods buyers, we run the following regression to assess whether experience goods buyers come directly to Amazon or through a search engine:

$$
\begin{aligned}
&\text{Logit}\left(\texttt{RefDomainIsAmazon}, \texttt{RefDomainIsSearchEngine}\right)_{i,t} \\
&= \tau_t \\
&+ \beta_1 \ \texttt{ExperienceGood}_{i,t} \\
&+ \epsilon_{i,t}
\end{aligned} \tag{6}
$$

Table 18: Product Characteristics Effects on Search Intensity for Amazon

	(1) RefDomainIsAmazon	(2) ReferringDomainIsSearchEngine
ExperienceGood	-4.274***	-0.828***
	(0.310)	(0.0672)
Observations	10791	10791

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Stata codes for generating Table 18 are listed below.

```
eststo: logit RefDomainIsAmazon ExperienceGood if domain_name == "amazon.com", vce(cluster MonthYear) noconstant

eststo: logit ReferringDomainIsSearchEngine ExperienceGood if domain_name == "amazon.com", vce(cluster MonthYear) noconstant

esttab using tableD3.tex, se label replace booktabs title(Product Characteristics Effects on Search Intensity for Amazon\label{tab:tabD3})

eststo clear
```

Code 19: Table D3 Generation

2.17 Table E1-E3

In this section, we examine the effect of a physical store closure on some other online retailers, to investigate concern that the effect of the Circuit City store may also be felt by the other smaller online consumer electronic stores.

2.17.1 Table E1

In order to check which online seller benefits due to the exit of the offline Circuit City store, we run the following regression model:

$$\begin{aligned}
&\log\left(\texttt{TotalMonthlySales}+1\right)_{i,t}\\
&=\mu_i+\tau_t\\
&+\beta_1\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\\
&+\beta_2\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\times\texttt{BBStorePresent}_i\\
&+\epsilon_{i,t}
\end{aligned}\tag{7}$$

Table 19: Results of the Sales Effect (All Product Categories)

	log(TotalMonthlySales + 1)			
	staples.com-0 Mile	walmart.com-0 Mile	dell.com-0 Mile	circuitcity.com-0 Mile
	(1)	(2)	(3)	(4)
β_1	−0.027	0.026	−0.006	0.003
	(0.064)	(0.018)	(0.018)	(0.036)
β_2	0.082	−0.034	−0.018	0.013
	(0.075)	(0.022)	(0.026)	(0.051)
Observations	8,352	24,912	19,440	3,940
R^2	0.0003	0.0001	0.0001	0.0001
Adjusted R^2	-0.046	-0.044	-0.045	-0.058
F Statistic	1.004 (df = 2; 7979)	1.332 (df = 2; 23849)	0.834 (df = 2; 18605)	0.094 (df = 2; 3722)
Note:				*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 19 are listed below.

```r
# Table E1
staple.tE1.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "staples.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
walmart.tE1.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "walmart.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
dell.tE1.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "dell.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
cc.tE1.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "circuitcity.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
```

Code 20: Table E1 Generation

2.17.2 Table E2

We run the following regression model to investigate if the change in search intensity as seen for amazon.com is also prevalent in the other major online seller categories.

$$
\begin{aligned}
&\log\left(\texttt{PagesPerDollar}+1,\texttt{MinsPerDollar}+1\right)_{i,t}\\
&=\mu_i+\tau_t\\
&+\beta_1\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\\
&+\beta_2\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\times\texttt{BBStorePresent}_i\\
&+\epsilon_{i,t}
\end{aligned}
\tag{8}
$$

The results of the search models are presented in Table 20 and 21.

Table 20: Results of the Online Search Effect (All Product Categories)

	log(PagesPerDollar + 1)			
	staples.com-0 Mile	walmart.com-0 Mile	dell.com-0 Mile	circuitcity.com-0 Mile
	(1)	(2)	(3)	(4)
β_1	0.010	0.004	0.001	−0.002
	(0.027)	(0.009)	(0.004)	(0.012)
β_2	−0.017	−0.002	−0.002	0.0002
	(0.031)	(0.011)	(0.005)	(0.016)
Observations	8,352	24,912	19,440	3,940
R^2	0.00004	0.00001	0.00001	0.00002
Adjusted R^2	-0.047	-0.045	-0.045	-0.058
F Statistic	0.171 (df = 2; 7979)	0.123 (df = 2; 23849)	0.083 (df = 2; 18605)	0.030 (df = 2; 3722)
Note:				*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 20 are listed below.

```
# Table E2
staple.tE2.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "staples.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
walmart.tE2.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "walmart.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
dell.tE2.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "dell.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
cc.tE2.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "circuitcity.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
```

Code 21: Table E2 Generation

2.17.3 Table E3

Table 21: Results of the Online Search Effect (All Product Categories)

	log(MinsPerDollar + 1)			
	staples.com-0 Mile	walmart.com-0 Mile	dell.com-0 Mile	circuitcity.com-0 Mile
	(1)	(2)	(3)	(4)
β_1	−0.011	0.002	−0.001	−0.002
	(0.022)	(0.008)	(0.003)	(0.010)
β_2	0.008	−0.001	−0.001	0.00003
	(0.027)	(0.010)	(0.005)	(0.014)
Observations	8,352	24,912	19,440	3,940
R^2	0.00003	0.00000	0.00001	0.00002
Adjusted R^2	-0.047	-0.045	-0.045	-0.058
F Statistic	0.137 (df = 2; 7979)	0.056 (df = 2; 23849)	0.124 (df = 2; 18605)	0.037 (df = 2; 3722)
Note:				*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 21 are listed below.

```
# Table E3
staple.tE3.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "staples.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
walmart.tE3.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "walmart.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
dell.tE3.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "dell.com",], index
      = c("Zip_Code", "Time"), model = "within", effect = "twoways")
cc.tE3.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_tE_balanced[data_0m_tE_balanced$domain_name == "circuitcity.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
```

Code 22: Table E3 Generation

2.18 Table G1-G3

2.18.1 Table G1

In order to further rule out other alternate explanations, we extracted the product categories most purchased by returning customers. We accordingly run the following regressions (with the selected product category sales as the outcome variable), for all the focal online competitors to Circuit City.

$$\begin{aligned}
&\log\left(\texttt{TotalMonthlySales}+1\right)_{i,t}\\
&=\mu_i+\tau_t\\
&+\beta_1\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\\
&+\beta_2\ \texttt{CCStorePresent}_i\times\texttt{AfterStoreClosing}_t\times\texttt{BBStorePresent}_i\\
&+\epsilon_{i,t}
\end{aligned}\tag{9}$$

Table 22: Results of the Sales Effect (Music, Movies and Videos, Console Video Games)

	log(TotalMonthlySales + 1)		
	amazon.com-0 Mile	bestbuy.com-0 Mile	circuitcity.com-0 Mile
	(1)	(2)	(3)
β_1	0.005	−0.001	−0.002
	(0.013)	(0.024)	(0.043)
β_2	0.008	0.000	
	(0.019)	(0.028)	
Observations	52,416	5,808	810
R^2	0.00002	0.00000	0.00000
Adjusted R^2	-0.044	-0.048	-0.092
F Statistic	0.535 (df = 2; 50207)	0.004 (df = 2; 5541)	0.001 (df = 1; 741)

Note: *p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 22 are listed below.

```r
# Table G1
ama.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "amazon.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
bb.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "bestbuy.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
stp.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "staples.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
walmart.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "dell.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
cc.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "circuitcity.com"
      ,], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
```

Code 23: Table G1 Generation

2.18.2 Table G2

We further, included all the zip codes in our dataset even if they did not have any sale at one of the five top online competitors. The results are given below:

Table 23: Results of the Sales Effect (All Products; All Online Sellers in the Control Group)

	log(TotalMonthlySales + 1)				
	amazon.com-0 Mile	bestbuy.com-0 Mile	staples.com-0 Mile	walmart.com-0 Mile	circuitcity.com-0 Mile
	(1)	(2)	(3)	(4)	(5)
β_1	0.014	−0.002	−0.027	−0.006	0.003
	(0.015)	(0.033)	(0.064)	(0.018)	(0.036)
β_2	−0.033	0.009	0.082	−0.018	0.013
	(0.022)	(0.036)	(0.075)	(0.026)	(0.051)
Observations	68,472	14,664	8,352	19,440	3,940
R^2	0.00003	0.00002	0.0003	0.0001	0.0001
Adjusted R^2	-0.044	-0.045	-0.046	-0.045	-0.058
F Statistic	1.091 (df = 2; 65594)	0.154 (df = 2; 14028)	1.004 (df = 2; 7979)	0.834 (df = 2; 18605)	0.094 (df = 2; 3722)

Note: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Codes for generating Table 23 are listed below.

```r
# Table G2
# Do not filter control group on target_domain_to_consider and no category filtering
ama.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "amazon.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
bb.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "bestbuy.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
stp.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "staples.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
walmart.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "dell.com",],
      index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
cc.tG.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_tG_balanced[data_0m_tG_balanced$domain_name == "circuitcity.com"
      ,], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
```

Code 24: Table G2 Generation

2.18.3 Table G3

There might also be a concern that we use individual level transactions to tease out how customers landed into Amazon's site after the store closure, not the zip code level aggregated data. Hence, we aggregated all the sales per zip code per month and calculated the ratio of sales navigation originating from a search engine and also those going directly to amazon.com. We then run a regression on both ratios as outcome variables.

$$\begin{aligned}
&(\texttt{AmazonReferringDomainIsSearchEngine Ratio}, \texttt{NoReferringDomain Ratio})_{i,t} \\
&= \mu_i + \tau_t \\
&+ \beta_1 \texttt{ CCStorePresent}_i \times \texttt{AfterStoreClosing}_t \\
&+ \beta_2 \texttt{ CCStorePresent}_i \times \texttt{AfterStoreClosing}_t \times \texttt{BBStorePresent}_i \\
&+ \epsilon_{i,t}
\end{aligned} \tag{10}$$

Table 24: Effect Referring Domain on Amazon Sales

	ReferringDomainIsSearchEngineRatio	NoReferringDomainRatio
	Amazon	Amazon
	(1)	(2)
β_1	-0.010^{**}	0.009^{*}
	(0.005)	(0.005)
β_2	0.011	-0.008
	(0.007)	(0.007)
Observations	73,416	73,416
R^2	0.0001	0.00004
Adjusted R^2	-0.044	-0.044
F Statistic (df = 2; 70332)	1.961	1.422
Note:		*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 24 are listed below.

```
# Table G3
ama.tG3.0mile.r1 <- plm(ReferringDomainIsSearchEngineRatio ~ DID + THREEINTER, data = data_0m_tG3_balanced[data_0m_tG3_balanced$domain_name == "
    amazon.com",], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
ama.tG3.0mile.r2 <- plm(NoReferringDomainRatio ~ DID + THREEINTER, data = data_0m_tG3_balanced[data_0m_tG3_balanced$domain_name == "amazon.com",],
    index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
```

Code 25: Table G3 Generation

3 Advanced Method

In this section, we present some advanced methods on the provided data.

3.1 Generalized Synthetic Control Method

In this dataset, the DID model is a good fit because 1) the treatment of all treated units begins at a same time t, which produces the parallel trend, and 2) no treatment heterogeneity exists. However, we still give a test to the GSC method. The GSC method merges the Interactive Fixed Effects (IFE) model and the Synthetic Control (SC) method. It accepts one treatment variable and observable control variables and handles unobserved factors in the model. R package `gsynth` is used in this section.

Because the three-way interaction variable `CCStorePresent` × `AfterStoreClosing` × `BBStorePresent` in the original paper is one of the way used to address endogeneity concern and the GSC model is able to handle the omitted variables problem and only accepts one treatment variable, we decide to focus on the interaction variable `CCStorePresent` × `AfterStoreClosing`. Also, because we are lacking BestBuy data regarding the pre-treatment periods and the number of units, the GSC model is not able to fit into Bestbuy data after grouping, which leads us to focus on Amazon data only.

Xu (2017) suggests that data should have at least 10 pre-treatment periods and 40 control units to apply the model. Since our data begins in January 2018 and the treatment begins in November 2018, the condition for the number of pre-treatment periods is satisfied. On the other hand, after grouping the data based on zip code and month, we have 2796 control units in the 0-mile data and 2855 control units in the 5-mile data. Therefore, the condition for the number of control variables is also satisfied.

R package `gsynth` allows us to feed in one outcome variable, one treatment variable and control variables and get the best value of r, the number of latent variables, directly after cross-validation. Besides, the model from the package drops the units which only have data records in less than n time periods, where n can be changes by users but should be at least 3. With larger n, the model involves less bias. However, because our data is imbalance and sparse, we are not able to use n larger than 3 because otherwise too many data will be dropped and the model cannot be applied. Also, we are restricted with r no greater than 1 because of the same reason. Figure 1 shows the panel view of a subset of data as an example. We can observe how sparse the data is and see the limited number of available periods in each units.

With all restrictions stated above, we run the model on sales effect and search breadth and depth on both 0-mile data and 5-mile data, with outcome variable `AmazonTotalMonthlySales`, `AmazonPagesPerDollar` and `AmazonMinsPerDollar`, respectively. Table 25 shows the chosen number of latent variables and the *Average Treatment effect on the Treated unit* (ATT) in each of the models.

Codes for each of the models are listed below.

```
################################## Amazon Total monthly sales 0 mile ####################################
# Select all the observable variables that can be grouped
# Group data by zip code and time
amazon_monthsales_0mile <- sqldf("SELECT SUM(prod_totprice) AS TotalMonthlySales, AVG(treatment) AS Treatment, AVG(BBStorePresent) AS
      BBStorePresent,
```

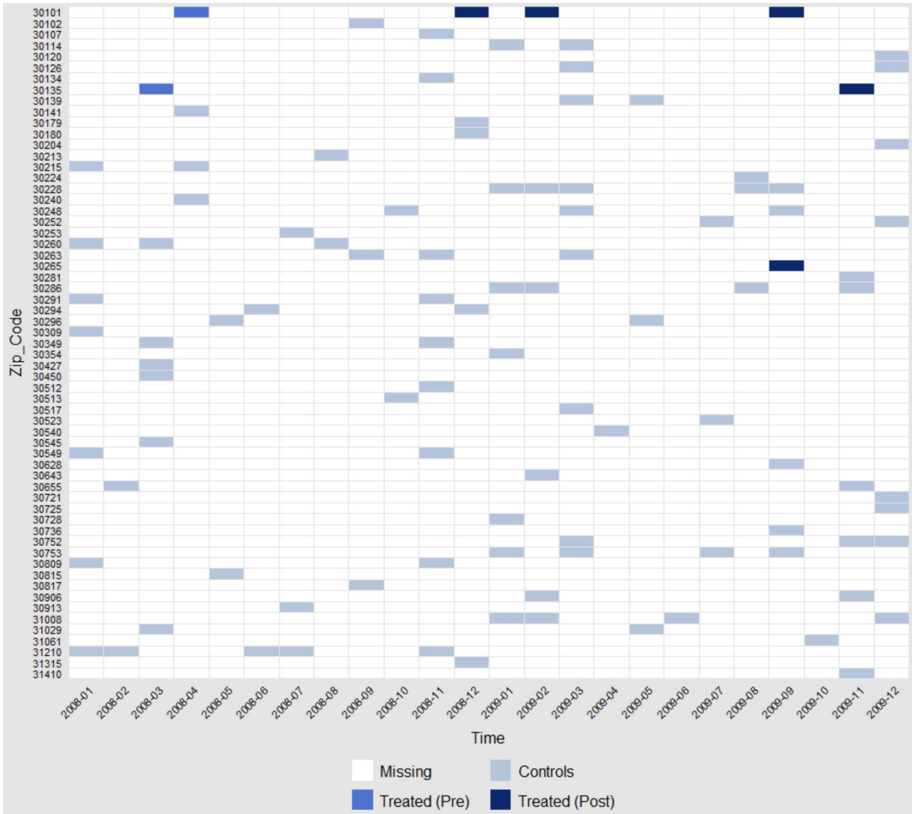

Figure 1: Panel View of a Subset of Amazon Data after Grouping

```
                              AVG(household_size) AS Household_Size, AVG(hoh_oldest_age) AS Hoh_Oldest_Age, AVG(household_income) AS Household_
      Income,
                              AVG(children) AS Children, AVG(connection_speed) AS Connection_Speed, MonthYear, Zip_Code
                              FROM concat_data1 WHERE domain_name='amazon.com' GROUP BY MonthYear, Zip_Code ORDER BY Zip_Code, MonthYear")
# number of control group = 2796
num_control = length(unique(amazon_monthsales_0mile$Zip_Code)) - length(unique(amazon_monthsales_0mile[which(amazon_monthsales_0mile$Treatment ==
      1), ]$Zip_Code))
# the last pre-treatment period = 10
t0 = length(unique(amazon_monthsales_0mile$MonthYear)) - length(unique(amazon_monthsales_0mile[which(amazon_monthsales_0mile$Treatment == 1), ]$
      MonthYear))

amazon_monthsales_0mile$logTotalMonthlySales = log(amazon_monthsales_0mile$TotalMonthlySales + 1)
amazon_monthsales_0mile$endo = amazon_monthsales_0mile$Treatment * amazon_monthsales_0mile$BBStorePresent

# Visualize the data structure for treated units and spot missing values
treated_zip = unique(amazon_monthsales_0mile[which(amazon_monthsales_0mile$Treatment == 1), ]$Zip_Code)
control_zip = unique(amazon_monthsales_0mile[which(amazon_monthsales_0mile$Treatment == 0), ]$Zip_Code)
panelView(logTotalMonthlySales ~ Treatment + endo + Household_Size + Hoh_Oldest_Age + Household_Income + Children + Connection_Speed,
          data = amazon_monthsales_0mile, show.id = c(970:1030), theme.bw = TRUE, index = c("Zip_Code","MonthYear"),
          xlab = "Time", axis.adjust = TRUE, pre.post = TRUE, main = "Panel View of a Subset of Amazon Data after Grouping")

                                                                                                                              but increase r
       (can only run with r={0,1} and min.To = 3)

# run the equation (can only run with r={0,1}.
out_amazon_monthsales_0mile <- gsynth(logTotalMonthlySales ~ Treatment + Household_Size + Hoh_Oldest_Age + Household_Income +
                Children + Connection_Speed, data = amazon_monthsales_0mile, index = c("Zip_Code","MonthYear"), force = "two-way",
              CV = TRUE, r = c(0,1), se = TRUE, inference = "parametric", min.T0 = 3, nboots = 1000, parallel = TRUE, seed = 1)

# insignificant result
print(out_amazon_monthsales_0mile)
```

Table 25: Number of Latent Variables and the Averaage Treatment Effect for Amazon Data

	AmazonTotalMonthlySales		AmazonPagesPerDollar		AmazonMinsPerDollar	
	Amazon-0 Mile	Amazon-5 Miles	Amazon-0 Mile	Amazon-5 Miles	Amazon-0 Mile	Amazon-5 Miles
	(1)	(2)	(3)	(4)	(5)	(6)
ATT	−0.863	−0.248	−0.210	0.024	−0.293	−0.293
	(1.316)	(0.229)	(18.580)	(0.211)	(1.038)	(1.067)
r*	0	0	1	0	0	0
MSPE of r	1.294	1.312	0.015	1.375	0.342	0.342

```r

# some figures
plot(out_amazon_monthsales_0mile, type = "raw", theme.bw = TRUE, axis.adjust = TRUE)
plot(out_amazon_monthsales_0mile, type = "counterfactual", raw = "all", theme.bw = TRUE, axis.adjust = TRUE)
################################ Amazon Total monthly sales 5 mile ####################################
# Select all the observable variables that can be grouped
# Group data by zip code and time
amazon_monthsales_5mile <- sqldf("SELECT SUM(prod_totprice) AS TotalMonthlySales, AVG(treatment) AS Treatment, AVG(BBStorePresent) AS BBStorePresent,
                                AVG(household_size) AS Household_Size, AVG(hoh_oldest_age) AS Hoh_Oldest_Age, AVG(household_income) AS Household_Income,
                                AVG(children) AS Children, AVG(connection_speed) AS Connection_Speed, MonthYear, Zip_Code
                                FROM concat_data2 WHERE domain_name='amazon.com' GROUP BY MonthYear, Zip_Code ORDER BY Zip_Code, MonthYear")
# number of control group = 2855
num_control = length(unique(amazon_monthsales_5mile$Zip_Code)) - length(unique(amazon_monthsales_5mile[which(amazon_monthsales_5mile$Treatment == 1), ]$Zip_Code))
# the last pre-treatment period = 10
t0 = length(unique(amazon_monthsales_5mile$MonthYear)) - length(unique(amazon_monthsales_5mile[which(amazon_monthsales_5mile$Treatment == 1), ]$MonthYear))

amazon_monthsales_5mile$logTotalMonthlySales = log(amazon_monthsales_5mile$TotalMonthlySales + 1)
amazon_monthsales_5mile$endo = amazon_mothsales_5mile$Treatment * amazon_monthsales_5mile$BBStorePresent

# run the equation (can only run with r={0,1}. MSPE increases as min.T0 increases)
out_amazon_monthsales_5mile <- gsynth(logTotalMonthlySales ~ Treatment + Household_Size + Hoh_Oldest_Age + Household_Income +
                                        Children + Connection_Speed, data = amazon_monthsales_5mile, index = c("Zip_Code","MonthYear"), force = "two-way",
                                      CV = TRUE, r = c(0,1), se = TRUE, inference = "parametric", min.T0 = 3, nboots = 1000, parallel = TRUE, seed = 1)

# insignificant result
print(out_amazon_monthsales_5mile)
################################ Amazon Pages Per Dollar 0 mile ####################################
# Select all the observable variables that can be grouped
# Group data by zip code and time
amazon_PagesPerDollar_0mile <- sqldf("SELECT PagesPerDollar, AVG(treatment) AS Treatment, AVG(BBStorePresent) AS BBStorePresent,
                                AVG(household_size) AS Household_Size, AVG(hoh_oldest_age) AS Hoh_Oldest_Age, AVG(household_income) AS Household_Income,
                                AVG(children) AS Children, AVG(connection_speed) AS Connection_Speed, MonthYear, Zip_Code
                                FROM concat_data1 WHERE domain_name='amazon.com' GROUP BY MonthYear, Zip_Code ORDER BY Zip_Code, MonthYear")

amazon_PagesPerDollar_0mile$logPagesPerDollar = log(amazon_PagesPerDollar_0mile$PagesPerDollar + 1)
amazon_PagesPerDollar_0mile$endo = amazon_PagesPerDollar_0mile$Treatment * amazon_PagesPerDollar_0mile$BBStorePresent

# run the equation (can only run with r={0,1} and min.To = 3)
out_amazon_PagesPerDollar_0mile <- gsynth(logPagesPerDollar ~ Treatment + Household_Size + Hoh_Oldest_Age + Household_Income +
                                      Children + Connection_Speed, data = amazon_PagesPerDollar_0mile, index = c("Zip_Code","MonthYear"), force = "two-way",
                                      CV = TRUE, r = c(0,1), se = TRUE, inference = "parametric", min.T0 = 3, nboots = 1000, parallel = TRUE, seed = 1)

```

```r
# insignificant result
print(out_amazon_PagesPerDollar_0mile)

# some figures
plot(out_amazon_PagesPerDollar_0mile, type = "raw", theme.bw = TRUE, axis.adjust = TRUE)
plot(out_amazon_PagesPerDollar_0mile, type = "counterfactual", raw = "all", theme.bw = TRUE, axis.adjust = TRUE)
################################ Amazon Pages Per Dollar 5 mile ####################################

amazon_PagesPerDollar_5mile <- sqldf("SELECT PagesPerDollar, AVG(treatment) AS Treatment, AVG(BBStorePresent) AS BBStorePresent,
                              AVG(household_size) AS Household_Size, AVG(hoh_oldest_age) AS Hoh_Oldest_Age, AVG(household_income) AS Household_Income,
                              AVG(children) AS Children, AVG(connection_speed) AS Connection_Speed, MonthYear, Zip_Code
                              FROM concat_data2 WHERE domain_name='amazon.com' GROUP BY MonthYear, Zip_Code ORDER BY Zip_Code, MonthYear")

amazon_PagesPerDollar_5mile$logPagesPerDollar = log(amazon_PagesPerDollar_5mile$PagesPerDollar + 1)
amazon_PagesPerDollar_5mile$endo = amazon_PagesPerDollar_5mile$Treatment * amazon_PagesPerDollar_5mile$BBStorePresent

# Without BBStorePresent, but increase r (can only run with r={0,1}). MSPE increases as min.T0 increases
out_amazon_PagesPerDollar_5mile <- gsynth(logPagesPerDollar ~ Treatment + Household_Size + Hoh_Oldest_Age + Household_Income +
                                        Children + Connection_Speed, data = amazon_PagesPerDollar_5mile, index = c("Zip_Code","MonthYear"), force = "two-way",
                                      CV = TRUE, r = c(0,1), se = TRUE, inference = "parametric", min.T0 = 3, nboots = 1000, parallel = TRUE, seed = 1)


# insignificant result
print(out_amazon_PagesPerDollar_5mile)
################################ Amazon Mins Per Dollar 0 mile ####################################

amazon_MinsPerDollar_0mile <- sqldf("SELECT MinsPerDollar, AVG(treatment) AS Treatment, AVG(BBStorePresent) AS BBStorePresent,
                              AVG(household_size) AS Household_Size, AVG(hoh_oldest_age) AS Hoh_Oldest_Age, AVG(household_income) AS Household_Income,
                              AVG(children) AS Children, AVG(connection_speed) AS Connection_Speed, MonthYear, Zip_Code
                              FROM concat_data1 WHERE domain_name='amazon.com' GROUP BY MonthYear, Zip_Code ORDER BY Zip_Code, MonthYear")

amazon_MinsPerDollar_0mile$logMinsPerDollar = log(amazon_MinsPerDollar_0mile$MinsPerDollar + 1)
amazon_MinsPerDollar_0mile$endo = amazon_MinsPerDollar_0mile$Treatment * amazon_MinsPerDollar_0mile$BBStorePresent

# run the equation (can only run  with r={0,1})
out_amazon_MinsPerDollar_0mile <- gsynth(logMinsPerDollar ~ Treatment + Household_Size + Hoh_Oldest_Age + Household_Income +
                                            Children + Connection_Speed, data = amazon_MinsPerDollar_0mile, index = c("Zip_Code","MonthYear"), force = "two-way",
                                          CV = TRUE, r = c(0,1), se = TRUE, inference = "parametric", min.T0 = 3, nboots = 1000, parallel = TRUE, seed = 1)

# insignificant result
print(out_amazon_MinsPerDollar_0mile)
################################ Amazon Mins Per Dollar 5 mile ####################################

amazon_MinsPerDollar_5mile <- sqldf("SELECT MinsPerDollar, AVG(treatment) AS Treatment, AVG(BBStorePresent) AS BBStorePresent,
                              AVG(household_size) AS Household_Size, AVG(hoh_oldest_age) AS Hoh_Oldest_Age, AVG(household_income) AS Household_Income,
                              AVG(children) AS Children, AVG(connection_speed) AS Connection_Speed, MonthYear, Zip_Code
                              FROM concat_data1 WHERE domain_name='amazon.com' GROUP BY MonthYear, Zip_Code ORDER BY Zip_Code, MonthYear")

amazon_MinsPerDollar_5mile$logMinsPerDollar = log(amazon_MinsPerDollar_0mile$MinsPerDollar + 1)
amazon_MinsPerDollar_5mile$endo = amazon_MinsPerDollar_5mile$Treatment * amazon_MinsPerDollar_5mile$BBStorePresent

# Without BBStorePresent, but increase r (can only run with r={0,1}). MSPE increases as min.T0 increases
out_amazon_MinsPerDollar_5mile <- gsynth(logMinsPerDollar ~ Treatment + Household_Size + Hoh_Oldest_Age + Household_Income +
                                           Children + Connection_Speed, data = amazon_MinsPerDollar_5mile, index = c("Zip_Code","MonthYear"), force = "two-way",
                                         CV = TRUE, r = c(0,1), se = TRUE, inference = "parametric", min.T0 = 3, nboots = 1000, parallel = TRUE, seed = 1)

# insignificant result
print(out_amazon_MinsPerDollar_5mile)
```

Code 26: Codes for GSC model

3.2 PSM

For robustness, we use nearest propensity score matching to match each zipcode from treatment group with an equivalent control group, using zip code level demographics. After matching, we have left with 89 matched zip codes. Below are the propensity score before and after matching.

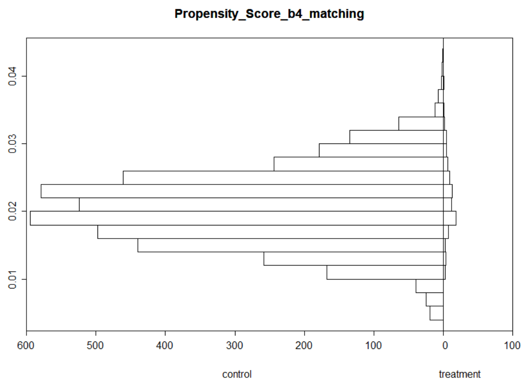

Figure 2: Scores Before Matching

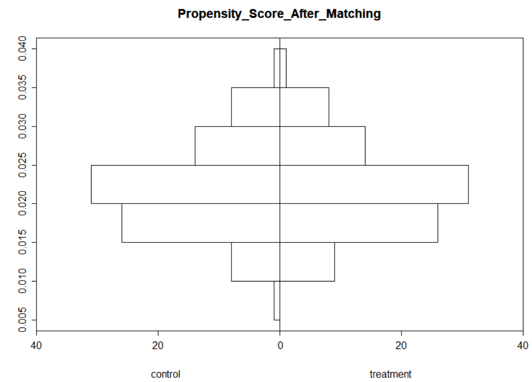

Figure 3: Scores After Matching

Table 26: Results of the Online Sales and Search Effect After Nearest Propensity Score Matching: TotalMonthlySales, PagesPerDollar, and MinsPerDollar (All Product Categories)

	log(TotalMonthlySales + 1)		log(PagesPerDollar + 1)		log(MinsPerDollar + 1)	
	Amazon-0 Mile	BesyBuy-0 Mile	Amazon-0 Mile	BesyBuy-0 Mile	Amazon-0 Mile	BesyBuy-0 Mile
	(1)	(2)	(3)	(4)	(5)	(6)
$beta_1$	0.029	0.0002	0.00004	0.0001	0.002	0.00000
	(0.024)	(0.031)	(0.020)	(0.009)	(0.019)	(0.005)
$beta_2$	−0.033	0.009	-0.068^{***}	0.003	-0.057^{***}	0.0004
	(0.028)	(0.031)	(0.023)	(0.009)	(0.022)	(0.005)
Observations	3,000	768	3,000	768	3,000	768
R^2	0.001	0.001	0.004	0.001	0.003	0.00004
Adjusted R^2	-0.052	-0.078	-0.048	-0.078	-0.049	-0.079
F Statistic	0.935 (df = 2; 2850)	0.192 (df = 2; 711)	6.219^{***} (df = 2; 2850)	0.185 (df = 2; 711)	4.693^{***} (df = 2; 2850)	0.014 (df = 2; 711)

Note: *p<0.1; **p<0.05; ***p<0.01

For this paper, look-ahead propensity score matching (LA-PSM) is not a viable method. The main assumption of LA-PSM is to match observations in treatment group with observations in control group that will convert to treatment group later. For example, if Circuit City would have closed 100 stores during November 2008 and another 100 store during November 2009, then we can be able to implement LA-PSM. However, in this case, all Circuit City stores were closed during at the same time, making LA-PSM impossible to implement.

The code for generating Table 26 are listed as below.

```
data_0m_psm_raw <- sqldf("SELECT Zip_Code, SUM(prod_totprice) AS TotalMonthlySales,
                     AVG(CCStorePresent) AS CCStorePresent,
                     AVG(household_size) AS HoHSize,
                     AVG(hoh_oldest_age) AS HoHAge,
                     AVG(household_income) AS HoHIncome,
                     AVG(children) AS HoHChildren,
                     AVG(connection_speed) AS HoHSpeed
                     FROM concat_data1 GROUP BY Zip_Code")

ps<- glm(CCStorePresent ~ HoHSize + HoHAge + HoHIncome + HoHChildren + HoHSpeed,
                                  data  =data_0m_psm_raw, family  = binomial())

summary(ps)

#Attach the predicted propensity  score to  the datafile
data_0m_psm_raw$psvalue <- predict(ps,  type  = "response")

#PSM  histogram
library(MatchIt)
library("RItools")
library(Hmisc)
histbackback(split(data_0m_psm_raw$psvalue, data_0m_psm_raw$CCStorePresent),
             main= "Propensity_Score_b4_matching",  xlab=c("control", "treatment"))

#---Match using near-neighbor
m.nn  <- matchit(CCStorePresent ~ HoHSize + HoHAge + HoHIncome + HoHChildren + HoHSpeed,
                   data =data_0m_psm_raw, method= "nearest",  ratio = 1)
summary(m.nn)
match_data  = match.data(m.nn)
plot(m.nn,  type  = "jitter")

histbackback(split(match_data$psvalue,match_data$CCStorePresent), main= "Propensity_Score_After_Matching",  xlab=c("control", "treatment"))


data_0m_psm <- sqldf("SELECT Zip_Code, MonthYear, domain_name, SUM(prod_totprice) AS TotalMonthlySales, SUM(pages_viewed) / SUM(prod_totprice) AS
      PagesPerDollar, SUM(duration) / SUM(prod_totprice) AS MinsPerDollar, AVG(CCStorePresent) AS CCStorePresent, AVG(BBStorePresent) AS
      BBStorePresent, AVG(AfterStoreClosing) AS AfterStoreClosing FROM concat_data1 GROUP BY Zip_Code, MonthYear, domain_name")
data_0m_psm_balanced <- dta_bal_imp_all(data_0m_psm)

# assign matched zipcode to dataset
data_0m_psm_balanced$Zipmatch <- ifelse(data_0m_psm_balanced$Zip_Code %in% match_data$Zip_Code, 1, 0)
data_0m_psm_balanced <- data_0m_psm_balanced[data_0m_psm_balanced$Zipmatch == 1, ]

# Table PSM
ama.psm.0mile.sale <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_psm_balanced[(data_0m_psm_balanced$domain_name == "amazon.
      com"),], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
ama.psm.pagesperdollar.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_psm_balanced[(data_0m_psm_balanced$domain_name == "
      amazon.com"),], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
ama.psm.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_psm_balanced[(data_0m_psm_balanced$domain_name == "
      amazon.com"),], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")

bb.psm.0mile.sale <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_psm_balanced[(data_0m_psm_balanced$domain_name == "bestbuy.
      com"),], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
bb.psm.pagesperdollar.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_psm_balanced[(data_0m_psm_balanced$domain_name == "
      bestbuy.com"),], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")
bb.psm.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_psm_balanced[(data_0m_psm_balanced$domain_name == "
      bestbuy.com"),], index = c("Zip_Code", "Time"), model = "within", effect = "twoways")

stargazer(ama.psm.0mile.sale, bb.psm.0mile.sale,
          ama.psm.pagesperdollar.0mile, bb.psm.pagesperdollar.0mile,
          ama.psm.minsperdollar.0mile, bb.psm.minsperdollar.0mile,
          title="Results of the Online Sales and Search Effect After Propensity Score Matching: TotalMonthlySales, PagesPerDollar, and
      MinsPerDollar (All Product Categories)",
          align=TRUE, covariate.labels=c("beta_1", "beta_2"), no.space=TRUE,
          column.sep.width = "1pt", label = "tab:tablepsm",
          column.labels=c("Amazon-0 Mile","BesyBuy-0 Mile","Amazon-0 Mile","BesyBuy-0 Mile","Amazon-0 Mile","BesyBuy-0 Mile"))
```

Code 27: Table PSM Generation

3.3 Causal Forest

In this section we focus on transaction level data and our treated group is determined by using a variable called `treatment` which takes value of 1 if `CCStorePresent` and `AfterStoreClosing` are 1 and takes value of 0 otherwise. We use the dataset called zero-mile data which only includes transactions from zip code areas where a Circuit City store was closed for our causal forest models. Codes for constructing treatment variable are listed below.

```
cf_d1 <- concat_data1 %>%
  mutate(treatment = ifelse(CCStorePresent==1 & AfterStoreClosing==1, 1, 0)) %>%
  select(-Store_Close_Status, -domain_id, -ref_domain_name, -MinsPerDollar,
         -event_date,-event_time,-tran_flg,-prod_name, -MonthYear,
         -CCStorePresent,- AfterStoreClosing,-BB_Store_Status, -PagesPerDollar,
         -site_session_id, -prod_category_id, -basket_tot, -machine_id, -Zip_Code)

cf_d2 <- concat_data2  %>%
  mutate(treatment=ifelse(CCStorePresent==1 & AfterStoreClosing==1, 1, 0)) %>%
  select(-Store_Close_Status,-domain_id, -ref_domain_name, -MinsPerDollar,
         -event_date,-event_time,-tran_flg,-prod_name, -MonthYear,
         -CCStorePresent, -AfterStoreClosing,-BB_Store_Status, -PagesPerDollar,
         -site_session_id,-prod_category_id, -basket_tot, -machine_id, -Zip_Code)

cf_all <- concat_all_data  %>%
  mutate(treatment=ifelse(CCStorePresent==1 & AfterStoreClosing==1, 1, 0)) %>%
  select(-Store_Close_Status,-domain_id, -ref_domain_name, -MinsPerDollar,
         -event_date,-event_time,-tran_flg,-prod_name, -MonthYear,
         -CCStorePresent, -AfterStoreClosing,-BB_Store_Status, -PagesPerDollar,
         -site_session_id,-prod_category_id, -basket_tot, -machine_id, -Zip_Code)
```

Code 28: Constructing Treatment Variable

We first investigate the treatment effects on Amazon sales using the transactions within the zip code where a Circuit City store was closed. For each transaction $i = 1, \cdots, n$, we observe a binary treatment indicator `treatment` (W_i), a real valued outcome log(`prod_totprice`) (Y_i), as well as 10 categorical covariates which are `hoh_most_education`, `census_region`, `household_size`, `hoh_oldest_age`, `children`, `racial_background`, `connection_speed`, `country_of_origin`, `prod_category_type` and `BBStorePresent`; and 4 real-valued covariates which are `pages_viewed`, `duration`, `prod_qty`, `household_income`. We expanded out categorical random variables via one-hot encoding, thus resulting in covariates $X_i \in \mathbb{R}^p$ with $p = 38$ or $p = 37$ depending on the response variable. We define causal effects via the potential outcomes model (Imbens and Rubin, 2015): For each sample i, the potential outcomes denoted by $Y_i(0)$ and $Y_i(1)$ corresponding to the outcome we would have observed if the i-th sample was in control or treatment group, and assume that we observe $Y_i = Y_i(Wi)$. The average treatment effect is then defined as $\tau = \mathbb{E}[Y_i(1) - Y_i(0)]$, and the conditional average treatment effect function is $\tau(x) = \mathbb{E}[Y_i(1) - Y_i(0) \mid X_i] = x$.
Codes for estimating treatment effects on the logarithm of Amazon sales within the zip code where a Circuit City store was closed are listed below.

```
## Amazon Sales Effect using Zero Mile Data
set.seed(1)

ama_cf_d1 <- cf_d1 %>%
  filter(domain_name=="amazon.com") %>%
  select(-domain_name)

W1_ama <- ama_cf_d1$treatment
```

```
Y1_ama <- ama_cf_d1$prod_totprice %>% log(.) %>% {.+1}

d1_ama <- ama_cf_d1 %>%
  select(-pages_viewed, -duration, -prod_qty,
         -prod_totprice, -household_income, -treatment)

d1_ama_exp <-model.matrix(~.+0, d1_ama)

X1_ama <- cbind(ama_cf_d1[,-c(4, 16, which(colnames(ama_cf_d1) %in% colnames(d1_ama)))], d1_ama_exp)

Y1_f_ama <- regression_forest(X1_ama, Y1_ama)
Y1_hat_ama <- predict(Y1_f_ama)$predictions

W1_f_ama <- regression_forest(X1_ama, W1_ama)
W1_hat_ama <- predict(W1_f_ama)$predictions

cf1_raw_ama <- causal_forest(X1_ama, Y1_ama, W1_ama,
                             Y.hat = Y1_hat_ama, W.hat = W1_hat_ama)

varimp1_ama <- variable_importance(cf1_raw_ama)
selected1_idx_ama <- which(varimp1_ama > mean(varimp1_ama))

cf1_ama <- causal_forest(X1_ama[,selected1_idx_ama], Y1_ama, W1_ama,
                         Y.hat = Y1_hat_ama, W.hat = W1_hat_ama,
                         tune.parameters = "all")

tau1_hat_ama <- predict(cf1_ama)$predictions
```

Code 29: Estimating Treatment Effects on the Logarithm of Amazon Sales (Zero Mile) with Causal Forests

We use the package **grf** (Tibshirani et al., 2018) to apply causal forest on our data and also to estimate the average treatment effect on treated (ATT). The confidence interval for the average treatment effect is presented in Table 27. Since the 90% confidence interval do not include zero, we can say that the average treatment effect on treated is negative and statistically significant at the 0.1 level. Since the average treatment effect on treated is negative and significant we can say that the closure of a a close by Circuit City store might have caused a decrease on Amazon sales.

Table 27: 90% CI for the ATT on log(`prod_totprice`) for amazon.com (Zero Mile Data)

5%	$\hat{\tau}_t$	95%
-0.42	-0.22	-0.03

As seen in Figure 4, the causal forest CATE estimates exhibit variation; but this does not necessarily imply that $\tau^{-i}(X_i)$ (The $^{(-i)}$ superscripts denote "out-of-bag" predictions) is a better estimate of $\tau(X_i)$ than the overall average treatment effect estimate $\hat{\tau}$ that we obtain using the doubly robust approach (The package **grf** uses this approach) (Athey and Wager, 2019). We try a test for heterogeneity, motivated by the "best linear predictor" method of Chernozhukov et al. (2018), that fits the CATE as a linear function of the the out-of-bag causal forest estimates $\hat{\tau}^{-i}(X_i)$. The results of this test is presented in Table 28. Since the differential forest prediction coefficient is significant and positive, we can say that the causal forest succeeded in finding heterogeneity. Next, we consider the effect of store closure on customers' online shopping behaviors. We run two different causal forest using dependent variables log(`PagesPerDollar`) which is the logarithm of the number of pages viewed for every dollar worth of products either on amazon.com or bestbuy.com and

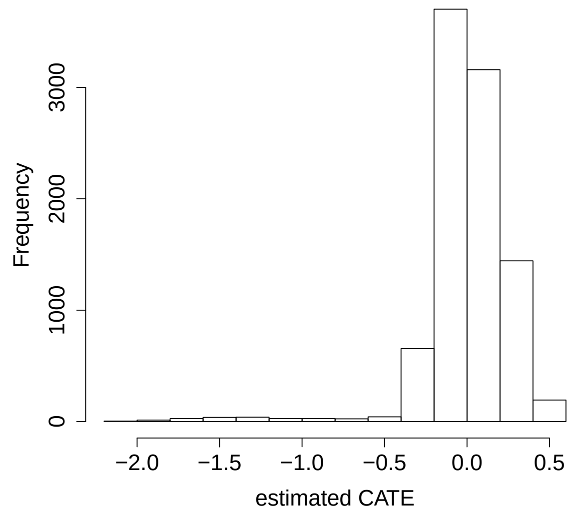

Figure 4: Histogram of out-of-bag estimates of CATE on logarithm of xAmazon Sales (Zero-Mile Data)

log(`MinsPerDollar`) which is the logarithm of the minutes spent at the website for every purchase dollar. We first investigate the treatment effect on log(`PagesPerDollar`) using the amazon.com transactions within the zip code where a Circuit City store was closed. The average treatment effect on treated and the 95% confidence interval is presented in Table (29) Since the 95% confidence interval do not include zero we can say that the average treatment effect on treated ($\hat{\tau}_t$) is positive and statistically significant at the 0.05 level. Looking at this result we can say that the closure of Circuit City stores seems to cause an increase on the number of pages that amazon customers observe per a dollar worth of purchase. This result is consistent with the findings of Samuel et al. (2020). Codes for estimating treatment effect on log(`PagesPerDollar`) using the transactions within the zip code where a Circuit City store was closed are listed below.

```
# Online Search Effect

## Search Breadth and Depth

### Amazon Pages Per Dollar using Zero Mile Data

cf_d3 <- concat_data1 %>%
  mutate(treatment=ifelse(CCStorePresent==1 & AfterStoreClosing==1, 1, 0)) %>%
  select(-Store_Close_Status, -domain_id, -ref_domain_name, -MinsPerDollar,
         -event_date,-event_time,-tran_flg,-prod_name, -MonthYear, -prod_totprice,
         -CCStorePresent,- AfterStoreClosing,-BB_Store_Status, -pages_viewed,
         -site_session_id, -prod_category_id, -basket_tot, -machine_id, -Zip_Code)

set.seed(1)

ama_cf_d3 <- cf_d3 %>%
  filter(domain_name=="amazon.com") %>%
  select(-domain_name)

W3_ama <- ama_cf_d3$treatment
Y3_ama <- ama_cf_d3$PagesPerDollar %>% log(.) %>% {.+1}

d3_ama <- ama_cf_d3 %>%
```

Table 28: Best linear fit using forest predictions for CATE on logarithm of Amazon Sales (Zero-mile dataset)

	CATE
mean.forest.prediction	4.950
	(4.259)
differential.forest.prediction	1.651***
	(0.506)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 29: 95% CI for the ATT on log(`PagesPerDollar`) for amazon.com (Zero Mile Data)

2.5%	$\hat{\tau}_t$	97.5%
0.02	0.27	0.52

```
  select(-duration, -prod_qty,
         -PagesPerDollar,
         -household_income,
         -treatment)

d3_ama_exp <-model.matrix(~.+0, d3_ama)

X3_ama <- cbind(ama_cf_d3[,-c(14, 15, which(colnames(ama_cf_d3) %in% colnames(d3_ama)))], d3_ama_exp)

Y3_f_ama <- regression_forest(X3_ama, Y3_ama)
Y3_hat_ama <- predict(Y3_f_ama)$predictions

W3_f_ama <- regression_forest(X3_ama, W3_ama)
W3_hat_ama <- predict(W3_f_ama)$predictions

cf3_raw_ama <- causal_forest(X3_ama, Y3_ama, W3_ama,
                             Y.hat = Y3_hat_ama, W.hat = W3_hat_ama)

varimp3_ama <- variable_importance(cf3_raw_ama)
selected3_idx_ama <- which(varimp3_ama > mean(varimp3_ama))

cf3_ama <- causal_forest(X3_ama[,selected3_idx_ama], Y3_ama, W3_ama,
                         Y.hat = Y3_hat_ama, W.hat = W3_hat_ama,
                         tune.parameters = "all")

tau3_hat_ama <- predict(cf3_ama)$predictions
```

Code 30: Estimating Treatment Effects on log(`PagesPerDollar`) for amazon.com (Zero Mile) with Causal Forests

We can see in Figure 5, the causal forest CATE estimates on log(`PagesPerDollar`) exhibit variation. We test for heterogeneity and report its the results in Table 30. We can say there is an evidence for heterogeneity

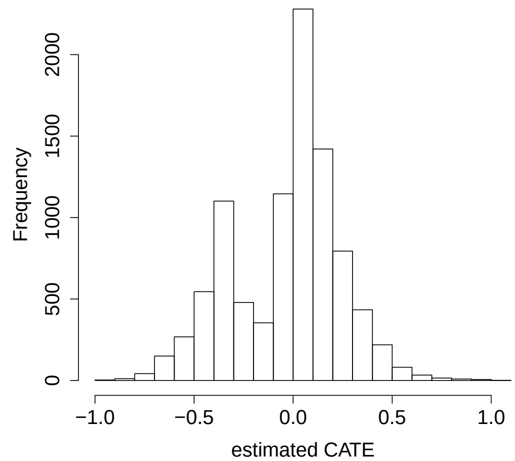

Figure 5: Histogram of out-of-bag estimates of CATE on log(`PagesPerDollar`) for amazon.com (Zero-Mile Data)

because the differential forest prediction coefficient is significant and positive.

Table 30: Best linear fit using forest predictions for CATE on `AmazonPagesPerDollar` (Zero-mile data)

	CATE
mean.forest.prediction	2.804
	(3.577)
differential.forest.prediction	1.288***
	(0.308)
Note:	*p<0.1; **p<0.05; ***p<0.01

Again to investigate the treatment effect on online search behaviors we use log(`MinsPerDollar`) using the amazon.com transactions within the zip code where a Circuit City store was closed. The average treatment effect on treated and the 90% confidence interval is presented in Table (31) Since the 90% confidence interval includes zero we fail to find an evidence for the significance of the average treatment effect. This result is different from what is shown in Samuel et al. (2020). We believe this comes from the fact that our dataset is actually a random subset of the original data that is used in the paper so our sample might not a good representation of that dataset. Since our estimate for the average treatment effect on treated is not significant we do not test the heterogeneity here but you can observe some heterogeneity in the estimated conditional average treatment effects from the Figure 5.

Table 31: 90% CI for the ATT on log(`MinsPerDollar`) for amazon.com (Zero Mile Data)

5%	$\hat{\tau}_t$	95%
-110.76	-24.32	62.13

Codes for estimating treatment effect on log(`MinsPerDollar`) using the transactions within the zip code where a Circuit City store was closed are listed below

```
### Amazon Minutes Per Dollar using Zero Mile Data

cf_d5 <- concat_data1 %>%
  mutate(treatment=ifelse(CCStorePresent==1 & AfterStoreClosing==1, 1, 0)) %>%
  select(-Store_Close_Status, -domain_id, -ref_domain_name, -PagesPerDollar,
         -event_date,-event_time,-tran_flg,-prod_name, -MonthYear, -prod_totprice,
         -CCStorePresent,- AfterStoreClosing,-BB_Store_Status, -duration,
         -site_session_id, -prod_category_id, -basket_tot, -machine_id, -Zip_Code)

set.seed(1)

ama_cf_d5 <- cf_d5 %>%
  filter(domain_name=="amazon.com") %>%
  select(-domain_name)

W5_ama <- ama_cf_d5$treatment
Y5_ama <- ama_cf_d5$MinsPerDollar %>% log(.) %>% {.+1}

d5_ama <- ama_cf_d5 %>%
  select(-pages_viewed,
         -prod_qty,
         -MinsPerDollar,
         -household_income,
          -treatment)

d5_ama_exp <-model.matrix(~.+0, d5_ama)

X5_ama <- cbind(ama_cf_d5[,-c(14, 15, which(colnames(ama_cf_d5) %in% colnames(d5_ama)))], d5_ama_exp)

Y5_f_ama <- regression_forest(X5_ama, Y5_ama)
Y5_hat_ama <- predict(Y5_f_ama)$predictions

W5_f_ama <- regression_forest(X5_ama, W5_ama)
W5_hat_ama <- predict(W5_f_ama)$predictions

cf5_raw_ama <- causal_forest(X5_ama, Y5_ama, W5_ama,
                             Y.hat = Y5_hat_ama, W.hat = W5_hat_ama)

varimp5_ama <- variable_importance(cf5_raw_ama)
selected5_idx_ama <- which(varimp5_ama > mean(varimp5_ama))

cf5_ama <- causal_forest(X5_ama[,selected5_idx_ama], Y5_ama, W5_ama,
                         Y.hat = Y5_hat_ama, W.hat = W5_hat_ama,
                         tune.parameters = "all")

tau5_hat_ama <- predict(cf5_ama)$predictions
```

Code 31: Estimating Treatment Effects on log(`PagesPerDollar`) for amazon.com (Zero Mile) with Causal Forests

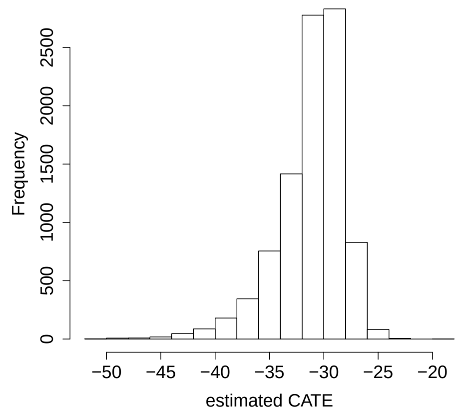

Figure 6: Histogram of out-of-bag estimates of CATE on log(`MinsPerDollar`) for amazon.com (Zero-Mile Data)

We do the same analyses using the Best Buy online transactions data but none of our casual forests estimations yields a significant results. We believe that since Best Buy owns physical stores, it didn't get affected by the closure of Circuit City stores as much as Amazon did. Our results for the transaction data from bestbuy.com is presented below.

Table 32: 90% CI for the ATT on log(`prod_totprice`) for bestbuy.com (Zero Mile Data)

5%	$\hat{\tau}_t$	95%
-0.39	0.08	0.55

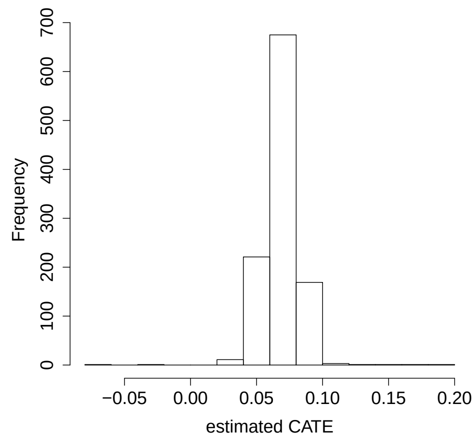

Figure 7: Histogram of out-of-bag estimates of CATE on logarithm of Best Buy Online Sales (Zero-Mile Data)

Table 33: 90% CI for the ATT on log(`PagesPerDollar`) for bestbuy.com (Zero Mile Data)

5%	$\hat{\tau}_t$	95%
-0.47	0.09	0.65

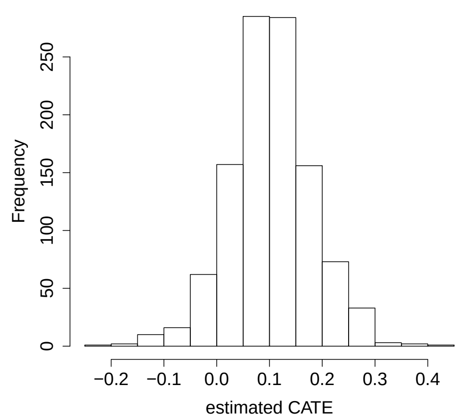

Figure 8: Histogram of out-of-bag estimates of CATE on log(`PagesPerDollar`) for bestbuy.com (Zero-Mile Data)

Table 34: 90% CI for the ATT on log(`MinsPerDollar`) for bestbuy.com (Zero Mile Data)

5%	$\hat{\tau}_t$	95%
-0.63	-0.29	0.05

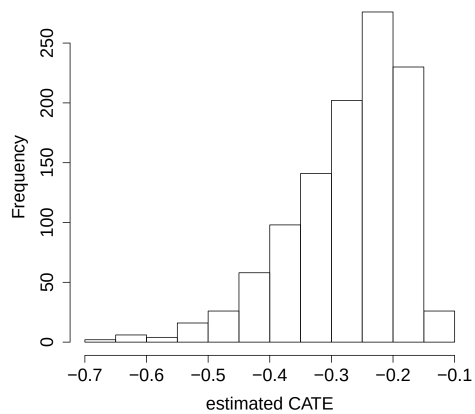

Figure 9: Histogram of out-of-bag estimates of CATE on log(`MinsPerDollar`) for bestbuy.com (Zero-Mile Data)

4 References

[1] Susan Athey and Stefan Wager. Estimating treatment effects with causal forests: An application. *arXiv preprint arXiv:1902.07409*, 2019.

[2] Badi H Baltagi and Seuck Heun Song. Unbalanced panel data: A survey. *Statistical Papers*, 47(4): 493–523, 2006.

[3] Victor Chernozhukov, Mert Demirer, Esther Duflo, and Ivan Fernandez-Val. Generic machine learning inference on heterogenous treatment effects in randomized experiments. Technical report, National Bureau of Economic Research, 2018.

[4] Yves Croissant, Giovanni Millo, et al. Panel data econometrics in r: The plm package. *Journal of statistical software*, 27(2):1–43, 2008.

[5] G Grothendieck. sqldf: manipulate r data frames using sql. *R package version 0.4-11*, 2017.

[6] Marek Hlavac. Stargazer: Well-formatted regression and summary statistics tables. *R package version*, 5(1), 2015.

[7] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences.* Cambridge University Press, 2015.

[8] Ben Jann. Estout: Stata module to export estimation results from estimates table. 2004.

[9] Allen McDowell. From the help desk: Seemingly unrelated regression with unbalanced equations. *The STATA journal*, 4(4):442–448, 2004.

[10] Jayarajan Samuel, Zhiqiang Eric Zheng, and Ying Xie. Value of local showrooms to online competitors. *MIS Quarterly*, 44(3), 2020.

[11] Julie Tibshirani, Susan Athey, Stefan Wager, Rina Friedberg, Luke Miner, Marvin Wright, Maintainer Julie Tibshirani, LinkingTo Rcpp, RcppEigen Imports DiceKriging, and GNU SystemRequirements. Package 'grf', 2018.

[12] Hadley Wickham and Evan Miller. haven: Import and export "spss","stata" and "sas" files. *R package version*, 1(0), 2018.

[13] Hadley Wickham, Romain Francois, Lionel Henry, Kirill Müller, et al. dplyr: A grammar of data manipulation. *R package version 0.4*, 3, 2015.

[14] Yiqing Xu. Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, 25(1):57–76, 2017.

[15] Achim Zeileis and Gabor Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *arXiv preprint math/0505527*, 2005.