

MIS7420

Seminar in Management Information Systems:

Paper Replication with R

Prof. Zhiqiang Zheng, Prof. Vijay Mookerjee

Author: Yihong Liu,
NETID: YXL180111,

Contents

List of Figures	3
List of Tables	4
List of Codes	5
1 Data Cleaning Process	6
2 Paper Replication	8
2.1 Table 1	8
2.2 Table 2	8
2.3 Table 3	9
2.4 Table 4	12
2.5 Table 5	13
2.6 Table 6	14
2.7 Table 7	15
2.8 Table 8	16
2.9 Table 9	17
2.10 Table 10	17
2.11 Table 11	17
2.12 Table 12	17
2.13 Table 13	17
2.14 Table 14	17
2.15 Table C1	17
2.16 Table D1-D4	18
2.17 Table E1-E2	18
2.18 Table G1-G3	18
3 Advanced Method	18
3.1 Synthetic Control Method	18
3.2 Generalized Synthetic Control Method	18
3.3 Heckit Approach	18
3.4 PSM and LA-PSM	18
3.5 Measurement Error Bias Correction	18
4 References	19

List of Figures

List of Tables

1	Summary Statistics of Top Five Vendors by Sales Volume	8
2	Summary Statistics of Referring Domain Categories	8
3	Average Difference-in-Difference (DID) of the Outcome Variables	9
4	Results of the Sales Effect (All Product Categories)	12
5	Results of the Search Effect (All Product Categories)	13
6	Results of the Sales Effect: Experience and Search Products	14
7	Results of the Online Search Effect: Experience Products	15
8	Results of the Online Search Effect: Search Products	16
9	Change in Demographics after Circuit City Store Closure	17

List of Codes

1	Data Preprocess	6
2	Table 1 Generation	8
3	Table 2 Generation	8
4	Table 3 Generation	9
5	Table 4 Generation	13
6	Table 5 Generation	13
7	Table 6 Generation	14
8	Table 7 Generation	15
9	Table 8 Generation	16

1 Data Cleaning Process

```

1 # load library
2 library('dplyr')
3 library('haven')
4 library('sqldf')
5 library('zoo')
6 library('plm')
7 library('stargazer')
8
9 # all data path
10 bb_zipcode_path <- 'data/bestbuyzipcodes_sample.sas7bdat'
11 sales_allother_zipcode_path <- 'data/sales_allotherzipcode_sample.sas7bdat'
12 sales_cc_0mile_path <- 'data/sales_cccity0milezipcode_sample.sas7bdat'
13 sales_cc_5miles_path <- 'data/sales_cccity5milezipcode_sample.sas7bdat'
14
15 # load data
16 bb_zipcode <- read_sas(bb_zipcode_path)
17 sales_allother_zipcode <- read_sas(sales_allother_zipcode_path)
18 sales_cc_0mile <- read_sas(sales_cc_0mile_path)
19 sales_cc_5miles <- read_sas(sales_cc_5miles_path)
20
21 # Data Mapping
22 sales_allother_zipcode$Store_Close_Status <- 0 # NaN means no CC in 5-miles radius, we change NaN to 0
23
24 # Exclude Data without purchase
25 # All data should be with purchase -> tran_flg == 1
26 sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$tran_flg == 1,]
27 sales_cc_0mile <- sales_cc_0mile[sales_cc_0mile$tran_flg == 1,]
28 sales_cc_5miles <- sales_cc_5miles[sales_cc_5miles$tran_flg == 1,]
29
30 # Filter Referring Domain
31
32 # groupby ref_domain and count
33 groupby_ref_domain_result <- aggregate(machine_id ~ ref_domain_name, rbind(sales_allother_zipcode, sales_cc_0mile, sales_cc_5miles), FUN = "length"
34 )
35 groupby_ref_domain_result <- groupby_ref_domain_result[order(-groupby_ref_domain_result$machine_id), ]
36 # we identify some search engines
37 search_engine_to_consider1 <- c("GOOGLE.COM", "YAHOO.COM", "google.com", "yahoo.com",
38 "MSN.COM", "msn.com", "aol.com", "AOL.COM", "LIVE.COM", "live.com",
39 "MYWEBSEARCH.COM", "ASK.COM", "MYWAY.COM", "mywebsearch.com",
40 "ask.com", "YAHOO.NET", "BIZRATE.COM", "bizrate.com",
41 "amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com",
42 "AMAZON.COM", "STAPLES.COM", "DELL.COM", "WALMART.COM", "BESTBUY.COM")
43 search_engine_to_consider2 <- c("GOOGLE.COM", "YAHOO.COM", "BING.COM", "google.com", "yahoo.com", "bing.com")
44
45 ref_domain_to_consider1 <- c("", "GOOGLE.COM", "YAHOO.COM", "google.com", "yahoo.com",
46 "MSN.COM", "msn.com", "aol.com", "AOL.COM", "LIVE.COM", "live.com",
47 "MYWEBSEARCH.COM", "ASK.COM", "MYWAY.COM", "mywebsearch.com",
48 "ask.com", "YAHOO.NET", "BIZRATE.COM", "bizrate.com",
49 "amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com",
50 "AMAZON.COM", "STAPLES.COM", "DELL.COM", "WALMART.COM", "BESTBUY.COM")
51
52 ref_domain_to_consider2 <- c("", "GOOGLE.COM", "YAHOO.COM", "BING.COM", "google.com", "yahoo.com", "bing.com")
53
54 # Then we filter data by refer domain name
55 sales_allother_zipcode <- sales_allother_zipcode[(sales_allother_zipcode$ref_domain_name %in% ref_domain_to_consider1),]
56 sales_cc_0mile <- sales_cc_0mile[(sales_cc_0mile$ref_domain_name %in% ref_domain_to_consider1),]
57 sales_cc_5miles <- sales_cc_5miles[(sales_cc_5miles$ref_domain_name %in% ref_domain_to_consider1),]
58
59 # Filter Target Domain Name
60 groupby_target_domain_result <- aggregate(machine_id ~ domain_name, rbind(sales_allother_zipcode, sales_cc_5miles), FUN = "length")
61 groupby_target_domain_result <- groupby_target_domain_result[order(-groupby_target_domain_result$machine_id), ]
62 five_target_domain_to_consider <- c("amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com")
63 two_target_domain_to_consider <- c("amazon.com", "bestbuy.com")
64
65 # we can choose what filter to apply
66 sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$domain_name %in% five_target_domain_to_consider,]
67 sales_cc_0mile <- sales_cc_0mile[sales_cc_0mile$domain_name %in% five_target_domain_to_consider,]
68 sales_cc_5miles <- sales_cc_5miles[sales_cc_5miles$domain_name %in% five_target_domain_to_consider,]

```

```

69
70 # Product Categories
71 # 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
72 # Jay removed 28, 30, 39, 40
73 # We choose to remove 38 39 40
74 sort(unique(rbind(sales_allother_zipcode, sales_cc_0mile, sales_cc_5miles)$prod_category_id))
75 category_to_consider <- c(22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37)
76 experience_product <- c(24, 25, 26, 27, 28, 31, 32, 33, 34, 36, 37)
77 search_product <- c(22, 23, 24, 29, 30, 35)
78
79 sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$prod_category_id %in% category_to_consider,]
80 sales_cc_0mile <- sales_cc_0mile[sales_cc_0mile$prod_category_id %in% category_to_consider,]
81 sales_cc_5miles <- sales_cc_5miles[sales_cc_5miles$prod_category_id %in% category_to_consider,]
82
83 # Date Transform
84 sales_allother_zipcode$event_date <- as.Date(sales_allother_zipcode$event_date)
85 sales_cc_0mile$event_date <- as.Date(sales_cc_0mile$event_date)
86 sales_cc_5miles$event_date <- as.Date(sales_cc_5miles$event_date)
87
88 # construct MonthYear - month of year
89 sales_allother_zipcode$MonthYear <- format(sales_allother_zipcode$event_date, "%Y-%m")
90 sales_cc_0mile$MonthYear <- format(sales_cc_0mile$event_date, "%Y-%m")
91 sales_cc_5miles$MonthYear <- format(sales_cc_5miles$event_date, "%Y-%m")
92
93 # Mark CC Closure
94
95 # CCStorePresent
96 # it is the same as Store_Close_Status
97 sales_allother_zipcode$CCStorePresent <- sales_allother_zipcode$Store_Close_Status
98 sales_cc_0mile$CCStorePresent <- sales_cc_0mile$Store_Close_Status
99 sales_cc_5miles$CCStorePresent <- sales_cc_5miles$Store_Close_Status
100
101 # AfterStoreClosing
102 sales_allother_zipcode$AfterStoreClosing <- ifelse(sales_allother_zipcode$MonthYear < "2008-11", 0, 1)
103 sales_cc_0mile$AfterStoreClosing <- ifelse(sales_cc_0mile$MonthYear < "2008-11", 0, 1)
104 sales_cc_5miles$AfterStoreClosing <- ifelse(sales_cc_5miles$MonthYear < "2008-11", 0, 1)
105
106 # BBStorePresent
107 sales_allother_zipcode <- merge(sales_allother_zipcode, bb_zipcode, by.x = "Zip_Code", by.y = "Zip_Code", all.x = TRUE)
108 sales_cc_0mile <- merge(sales_cc_0mile, bb_zipcode, by.x = "Zip_Code", by.y = "Zip_Code", all.x = TRUE)
109 sales_cc_5miles <- merge(sales_cc_5miles, bb_zipcode, by.x = "Zip_Code", by.y = "Zip_Code", all.x = TRUE)
110
111 sales_allother_zipcode$BBStorePresent <- na.fill(sales_allother_zipcode$BB_Store_Status, 0)
112 sales_cc_0mile$BBStorePresent <- na.fill(sales_cc_0mile$BB_Store_Status, 0)
113 sales_cc_5miles$BBStorePresent <- na.fill(sales_cc_5miles$BB_Store_Status, 0)
114
115 # Mark Referring Domain
116 # Question: How to group data?
117 sales_allother_zipcode$NoReferringDomain <- ifelse(sales_allother_zipcode$ref_domain_name == "", 1, 0)
118 sales_cc_0mile$NoReferringDomain <- ifelse(sales_cc_0mile$ref_domain_name == "", 1, 0)
119 sales_cc_5miles$NoReferringDomain <- ifelse(sales_cc_5miles$ref_domain_name == "", 1, 0)
120
121 sales_allother_zipcode$ReferringDomainIsSearchEngine <- ifelse(sales_allother_zipcode$ref_domain_name %in% search_engine_to_consider1, 1, 0)
122 sales_cc_0mile$ReferringDomainIsSearchEngine <- ifelse(sales_cc_0mile$ref_domain_name %in% search_engine_to_consider1, 1, 0)
123 sales_cc_5miles$ReferringDomainIsSearchEngine <- ifelse(sales_cc_5miles$ref_domain_name %in% search_engine_to_consider1, 1, 0)
124
125 # Aggregate Data
126 concat_data1 <- rbind(sales_allother_zipcode, sales_cc_0mile)
127 concat_data2 <- rbind(sales_allother_zipcode, sales_cc_5miles)
128 concat_data1_exp <- concat_data1[concat_data1$prod_category_id %in% experience_product, ]
129 concat_data1_search <- concat_data1[concat_data1$prod_category_id %in% search_product, ]
130 concat_data2_exp <- concat_data2[concat_data2$prod_category_id %in% experience_product, ]
131 concat_data2_search <- concat_data2[concat_data2$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.

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

DomainName	TotalTransaction	TotalSales	TotalPagesViewed	PagesPerDollar	TotalDuration	MinsPerDollar
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.

```
1 # Table 1
2 table1_raw <- rbind(read_sas(sales_allother_zipcode_path), read_sas(sales_cc_0mile_path))
3 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")
4 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 Transaction	Referred by SearchEngine	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.

```
1 # Table 2
2
3 table2_raw <- rbind(read_sas(sales_allother_zipcode_path), read_sas(sales_cc_0mile_path))
4 table2_raw$direct_to_website <- ifelse(table2_raw$ref_domain_name == '', 1, 0)
5 table2_raw$referred_by_search <- ifelse(table2_raw$ref_domain_name %in% search_engine_to_consider1, 1, 0)
6 table2_raw$referred_by_other <- ifelse(!(table2_raw$ref_domain_name %in% ref_domain_to_consider1), 1, 0)
```



```
7 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.

```
1 # Table 3
2 temp <- read_sas(sales_allother_zipcode_path)
3 temp$Store_Close_Status <- 0
4 table3_0m_raw <- rbind(temp, read_sas(sales_cc_0mile_path))
5 table3_5m_raw <- rbind(temp, read_sas(sales_cc_5miles_path))
6
7 # Date Transform
8 table3_0m_raw$event_date <- as.Date(table3_0m_raw$event_date)
9 table3_5m_raw$event_date <- as.Date(table3_5m_raw$event_date)
10
```

```

11 # construct MonthYear - month of year
12 table3_0m_raw$MonthYear <- format(table3_0m_raw$event_date, "%Y-%m")
13 table3_5m_raw$MonthYear <- format(table3_5m_raw$event_date, "%Y-%m")
14
15 # Mark CC Closure
16
17 # CCStorePresent
18 # it is the same as Store_Close_Status
19 table3_0m_raw$CCStorePresent <- table3_0m_raw$Store_Close_Status
20 table3_5m_raw$CCStorePresent <- table3_5m_raw$Store_Close_Status
21
22 # AfterStoreClosing
23 table3_0m_raw$AfterStoreClosing <- ifelse(table3_0m_raw$MonthYear < "2008-11", 0, 1)
24 table3_5m_raw$AfterStoreClosing <- ifelse(table3_5m_raw$MonthYear < "2008-11", 0, 1)
25
26 # BBStorePresent
27 table3_0m_raw <- merge(table3_0m_raw, bb_zipcode, by.x = "Zip_Code", by.y = "Zip_Code", all.x = TRUE)
28 table3_5m_raw <- merge(table3_5m_raw, bb_zipcode, by.x = "Zip_Code", by.y = "Zip_Code", all.x = TRUE)
29
30 table3_0m_raw$BBStorePresent <- na.fill(table3_0m_raw$BB_Store_Status, 0)
31 table3_5m_raw$BBStorePresent <- na.fill(table3_5m_raw$BB_Store_Status, 0)
32
33 # aggregate data
34
35 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")
36 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")
37
38 # Table 3 Gen Func
39 table3_gen <- function(table3_raw, domain_name_used, print_name){
40   # Amazon Sales
41   # for control
42   amazonsales_control_before <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
    AfterStoreClosing == 0),]$TotalMonthlySales
43   amazonsales_control_after <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
    AfterStoreClosing == 1),]$TotalMonthlySales
44
45   amazonsales_control_before <- log(amazonsales_control_before + 1)
46   amazonsales_control_after <- log(amazonsales_control_after + 1)
47   # t test
48   t_test.amazonsales_control <- t.test(amazonsales_control_after, amazonsales_control_before)
49   amazonsales_control_mean_diff_se <- t_test.amazonsales_control$stderr
50   t_test.amazonsales_control$p.value
51   amazonsales_control_after_mean <- t_test.amazonsales_control$estimate[["mean of x"]]
52   amazonsales_control_before_mean <- t_test.amazonsales_control$estimate[["mean of y"]]
53   amazonsales_control_mean_diff <- t_test.amazonsales_control$estimate[["mean of x"]] - t_test.amazonsales_control$estimate[["mean of y"]]
54
55   # Amazon Sales
56   # for treatment
57   amazonsales_treatment_before <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
    AfterStoreClosing == 0),]$TotalMonthlySales
58   amazonsales_treatment_after <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
    AfterStoreClosing == 1),]$TotalMonthlySales
59
60   amazonsales_treatment_before <- log(amazonsales_treatment_before + 1)
61   amazonsales_treatment_after <- log(amazonsales_treatment_after + 1)
62   # t test
63   t_test.amazonsales_treatment <- t.test(amazonsales_treatment_after, amazonsales_treatment_before)
64   amazonsales_treatment_mean_diff_se <- t_test.amazonsales_treatment$stderr
65   t_test.amazonsales_treatment$p.value
66   amazonsales_treatment_after_mean <- t_test.amazonsales_treatment$estimate[["mean of x"]]
67   amazonsales_treatment_before_mean <- t_test.amazonsales_treatment$estimate[["mean of y"]]
68   amazonsales_treatment_mean_diff <- t_test.amazonsales_treatment$estimate[["mean of x"]] - t_test.amazonsales_treatment$estimate[["mean of y"]]
69
70   # Amazon Sales DID
71   amazonsales_did <- amazonsales_treatment_mean_diff - amazonsales_control_mean_diff
72

```

```

73 # Amazon PagesPerDollar
74 # for control
75 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),]$TotalMonthlySales
76 amazonppd_control_after <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
  AfterStoreClosing == 1),]$TotalPages / table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_
  raw$AfterStoreClosing == 1),]$TotalMonthlySales
77
78 amazonppd_control_before <- log(amazonppd_control_before + 1)
79 amazonppd_control_after <- log(amazonppd_control_after + 1)
80 # t test
81 t_test.amazonppd_control <- t.test(amazonppd_control_after, amazonppd_control_before)
82 amazonppd_control_mean_diff_se <- t_test.amazonppd_control$stderr
83 t_test.amazonppd_control$p.value
84 amazonppd_control_after_mean <- t_test.amazonppd_control$estimate[["mean of x"]]
85 amazonppd_control_before_mean <- t_test.amazonppd_control$estimate[["mean of y"]]
86 amazonppd_control_mean_diff <- t_test.amazonppd_control$estimate[["mean of x"]] - t_test.amazonppd_control$estimate[["mean of y"]]
87
88 # Amazon PagesPerDollar
89 # for treatment
90 amazonppd_treatment_before <- table3_raw[(table3_raw$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),]$TotalMonthlySales
91 amazonppd_treatment_after <- table3_raw[(table3_raw$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),]$TotalMonthlySales
92
93 amazonppd_treatment_before <- log(amazonppd_treatment_before + 1)
94 amazonppd_treatment_after <- log(amazonppd_treatment_after + 1)
95 # t test
96 t_test.amazonppd_treatment <- t.test(amazonppd_treatment_after, amazonppd_treatment_before)
97 amazonppd_treatment_mean_diff_se <- t_test.amazonppd_treatment$stderr
98 t_test.amazonppd_treatment$p.value
99 amazonppd_treatment_after_mean <- t_test.amazonppd_treatment$estimate[["mean of x"]]
100 amazonppd_treatment_before_mean <- t_test.amazonppd_treatment$estimate[["mean of y"]]
101 amazonppd_treatment_mean_diff <- t_test.amazonppd_treatment$estimate[["mean of x"]] - t_test.amazonppd_treatment$estimate[["mean of y"]]
102
103 # Amazon PagesPerDollar DID
104 amazonppd_did <- amazonppd_treatment_mean_diff - amazonppd_control_mean_diff
105
106 # Amazon MinsPerDollar
107 # for control
108 amazonmpd_control_before <- table3_raw[(table3_raw$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),]$TotalMonthlySales
109 amazonmpd_control_after <- table3_raw[(table3_raw$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),]$TotalMonthlySales
110
111 amazonmpd_control_before <- log(amazonmpd_control_before + 1)
112 amazonmpd_control_after <- log(amazonmpd_control_after + 1)
113 # t test
114 t_test.amazonmpd_control <- t.test(amazonmpd_control_after, amazonmpd_control_before)
115 amazonmpd_control_mean_diff_se <- t_test.amazonmpd_control$stderr
116 t_test.amazonmpd_control$p.value
117 amazonmpd_control_after_mean <- t_test.amazonmpd_control$estimate[["mean of x"]]
118 amazonmpd_control_before_mean <- t_test.amazonmpd_control$estimate[["mean of y"]]
119 amazonmpd_control_mean_diff <- t_test.amazonmpd_control$estimate[["mean of x"]] - t_test.amazonmpd_control$estimate[["mean of y"]]
120
121 # Amazon MinsPerDollar
122 # for treatment
123 amazonmpd_treatment_before <- table3_raw[(table3_raw$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),]$TotalMonthlySales
124 amazonmpd_treatment_after <- table3_raw[(table3_raw$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
125
126 amazonmpd_treatment_before <- log(amazonmpd_treatment_before + 1)
127 amazonmpd_treatment_after <- log(amazonmpd_treatment_after + 1)
128 # t test

```

```

129 t_test.amazonmpd_treatment <- t.test(amazonmpd_treatment_after, amazonmpd_treatment_before)
130 amazonmpd_treatment_mean_diff_se <- t_test.amazonmpd_treatment$stderr
131 t_test.amazonmpd_treatment$p.value
132 amazonmpd_treatment_after_mean <- t_test.amazonmpd_treatment$estimate[["mean of x"]]
133 amazonmpd_treatment_before_mean <- t_test.amazonmpd_treatment$estimate[["mean of y"]]
134 amazonmpd_treatment_mean_diff <- t_test.amazonmpd_treatment$estimate[["mean of x"]] - t_test.amazonmpd_treatment$estimate[["mean of y"]]
135
136 # Amazon MinsPerDollar DID
137 amazonmpd_did <- amazonmpd_treatment_mean_diff - amazonmpd_control_mean_diff
138
139 # construct table
140 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),
141             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),
142             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),
143             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),
144             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),
145             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))
146 )
147 }
148
149 # generate table
150 amazon_table3 <- table3_gen(table3_0m_aggregate, "amazon.com", "Amazon")
151 bestbuy_table3 <- table3_gen(table3_0m_aggregate, "bestbuy.com", "bestbuy.com")
152
153 #
154 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

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	0.138 (0.524)	-0.202 (0.198)	0.090 (1.171)	-1.002 (0.896)
β_2	0.109 (0.668)	0.096 (0.245)		1.116 (1.224)
Observations	5,126	5,781	729	837
R ²	0.0002	0.001	0.0001	0.012
Adjusted R ²	-1.279	-1.199	-6.744	-6.512
F Statistic	0.204 (df = 2; 2248)	0.672 (df = 2; 2627)	0.006 (df = 1; 94)	0.647 (df = 2; 110)

Note: Standard errors are in parentheses

*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 4 are listed below.

```

1 # Table 4 Data
2 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")
3 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")
4 # manually construct DID and THREEINTERACTION
5 data_0m_t4$DID <- data_0m_t4$CCStorePresent * data_0m_t4$AfterStoreClosing
6 data_0m_t4$THREEINTER <- data_0m_t4$CCStorePresent * data_0m_t4$AfterStoreClosing * data_0m_t4$BBStorePresent
7 data_5m_t4$DID <- data_5m_t4$CCStorePresent * data_5m_t4$AfterStoreClosing
8 data_5m_t4$THREEINTER <- data_5m_t4$CCStorePresent * data_5m_t4$AfterStoreClosing * data_5m_t4$BBStorePresent
9 # Table 4
10 ama.t4.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t4[data_0m_t4$domain_name == "amazon.com",], index = c("Zip_Code"
  , "MonthYear"), model = "within", effect = "twoways")
11 ama.t4.5mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t4[data_5m_t4$domain_name == "amazon.com",], index = c("Zip_Code"
  , "MonthYear"), model = "within", effect = "twoways")
12 bb.t4.0mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t4[data_0m_t4$domain_name == "bestbuy.com",], index = c("Zip_Code"
  , "MonthYear"), model = "within", effect = "twoways")
13 bb.t4.5mile <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t4[data_5m_t4$domain_name == "bestbuy.com",], index = c("Zip_Code"
  , "MonthYear"), model = "within", effect = "twoways")

```

Code 5: Table 4 Generation

2.5 Table 5

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	-0.078 (0.411)	0.072 (0.156)	0.476 (0.680)	0.373 (0.511)	0.149 (0.400)	0.063 (0.152)	0.127 (0.544)	0.513 (0.425)
β_2	-0.263 (0.524)	-0.056 (0.193)		-0.028 (0.697)	-0.387 (0.510)	-0.047 (0.188)		-0.309 (0.580)
Observations	5,126	5,781	729	837	5,126	5,781	729	837
R ²	0.0005	0.0001	0.005	0.008	0.0003	0.0001	0.001	0.014
Adjusted R ²	-1.279	-1.200	-6.705	-6.537	-1.279	-1.200	-6.740	-6.490
F Statistic	0.552 (df = 2; 2248)	0.109 (df = 2; 2627)	0.490 (df = 1; 94)	0.458 (df = 2; 110)	0.347 (df = 2; 2248)	0.089 (df = 2; 2627)	0.055 (df = 1; 94)	0.805 (df = 2; 110)

Note:

*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 5 are listed below.

```

1 # Table 5 Data
2 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")
3 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")
4 # manually construct DID and THREEINTERACTION
5 data_0m_t5$DID <- data_0m_t5$CCStorePresent * data_0m_t5$AfterStoreClosing
6 data_0m_t5$THREEINTER <- data_0m_t5$CCStorePresent * data_0m_t5$AfterStoreClosing * data_0m_t5$BBStorePresent
7 data_5m_t5$DID <- data_5m_t5$CCStorePresent * data_5m_t5$AfterStoreClosing
8 data_5m_t5$THREEINTER <- data_5m_t5$CCStorePresent * data_5m_t5$AfterStoreClosing * data_5m_t5$BBStorePresent
9 # Table 5
10 # For PagesPerDollar
11 ama.t5.pagesperdollar.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "amazon.com",], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
12 ama.t5.pagesperdollar.5mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "amazon.com",], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")

```

```

13 bb.t5.pagesperdollar.0mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "bestbuy.com",], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
14 bb.t5.pagesperdollar.5mile <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "bestbuy.com",], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
15 # For MinsPerDollar
16 ama.t5.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "amazon.com",], index = c(
  "Zip_Code", "MonthYear"), model = "within", effect = "twoways")
17 ama.t5.minsperdollar.5mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "amazon.com",], index = c(
  "Zip_Code", "MonthYear"), model = "within", effect = "twoways")
18 bb.t5.minsperdollar.0mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t5[data_0m_t5$domain_name == "bestbuy.com",], index = c(
  "Zip_Code", "MonthYear"), model = "within", effect = "twoways")
19 bb.t5.minsperdollar.5mile <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t5[data_5m_t5$domain_name == "bestbuy.com",], index = c(
  "Zip_Code", "MonthYear"), model = "within", effect = "twoways")

```

Code 6: Table 5 Generation

2.6 Table 6

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

	log(TotalMonthlySales + 1)						
	Amazon 0 Mile Experience	Amazon 5 Miles Experience	Amazon 0 Mile Search	Amazon 5 Miles Search	BestBuy 0 Mile Experience	BestBuy 5 Miles Experience	BestBuy 5 Miles Search
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
β_1	-1.783 (1.821)	0.245 (0.697)	0.517 (0.634)	-0.304* (0.180)	-1.574 (2.994)	-2.021 (2.054)	-2.515** (0.993)
β_2	1.785 (2.015)	-0.173 (0.772)	-0.069 (0.816)	0.281 (0.229)		2.867 (2.277)	
Observations	1,788	2,022	3,741	4,229	461	529	342
R ²	0.002	0.0003	0.001	0.002	0.024	0.090	0.130
Adjusted R ²	-3.195	-2.923	-1.437	-1.343	-39.794	-29.016	-5.901
F Statistic	0.480 (df = 2; 425)	0.071 (df = 2; 515)	0.704 (df = 2; 1533)	1.424 (df = 2; 1802)	0.276 (df = 1; 11)	0.795 (df = 2; 16)	6.417** (df = 1; 43)

Note:

*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 6 are listed below.

```

1 # Table 6 Data
2 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")
3 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")
4 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")
5 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")
6 # manually construct DID and THREEINTERACTION
7 data_0m_t6_exp$DID <- data_0m_t6_exp$CCStorePresent * data_0m_t6_exp$AfterStoreClosing
8 data_0m_t6_exp$THREEINTER <- data_0m_t6_exp$CCStorePresent * data_0m_t6_exp$AfterStoreClosing * data_0m_t6_exp$BBStorePresent
9 data_0m_t6_search$DID <- data_0m_t6_search$CCStorePresent * data_0m_t6_search$AfterStoreClosing
10 data_0m_t6_search$THREEINTER <- data_0m_t6_search$CCStorePresent * data_0m_t6_search$AfterStoreClosing * data_0m_t6_search$BBStorePresent
11 data_5m_t6_exp$DID <- data_5m_t6_exp$CCStorePresent * data_5m_t6_exp$AfterStoreClosing
12 data_5m_t6_exp$THREEINTER <- data_5m_t6_exp$CCStorePresent * data_5m_t6_exp$AfterStoreClosing * data_5m_t6_exp$BBStorePresent
13 data_5m_t6_search$DID <- data_5m_t6_search$CCStorePresent * data_5m_t6_search$AfterStoreClosing

```

```

14 data_5m_t6_search$THREEINTER <- data_5m_t6_search$CCStorePresent * data_5m_t6_search$AfterStoreClosing * data_5m_t6_search$BBStorePresent
15 # Table 6
16 # AmazonTotalMonthlySales & BBTotalMonthlySale vs Experience and Search Product
17 ama.t6.0mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_exp[data_0m_t6_exp$domain_name == "amazon.com"], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
18 ama.t6.5mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_exp[data_5m_t6_exp$domain_name == "amazon.com"], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
19 ama.t6.0mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_search[data_0m_t6_search$domain_name == "amazon.com"],
  index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
20 ama.t6.5mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_search[data_5m_t6_search$domain_name == "amazon.com"],
  index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
21 bb.t6.0mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_exp[data_0m_t6_exp$domain_name == "bestbuy.com"], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
22 bb.t6.5mile.exp <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_exp[data_5m_t6_exp$domain_name == "bestbuy.com"], index =
  c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
23 bb.t6.0mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_0m_t6_search[data_0m_t6_search$domain_name == "bestbuy.com"],
  index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
24 bb.t6.5mile.search <- plm(log(TotalMonthlySales + 1) ~ DID + THREEINTER, data = data_5m_t6_search[data_5m_t6_search$domain_name == "bestbuy.com"],
  index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")

```

Code 7: Table 6 Generation

2.7 Table 7

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	0.663 (1.451)	-0.067 (0.547)	1.091 (0.800)	0.136 (0.634)	1.046 (1.406)	0.039 (0.536)	0.886 (0.625)	0.386 (0.481)
β_2	-1.167 (1.605)	0.145 (0.606)		0.043 (0.703)	-1.366 (1.555)	-0.021 (0.593)		-0.122 (0.533)
Observations	1,788	2,022	461	529	1,788	2,022	461	529
R ²	0.002	0.0002	0.145	0.013	0.002	0.00002	0.155	0.069
Adjusted R ²	-3.198	-2.924	-34.767	-31.568	-3.197	-2.924	-34.355	-29.730
F Statistic	0.358 (df = 2; 425)	0.040 (df = 2; 515)	1.861 (df = 1; 11)	0.106 (df = 2; 16)	0.390 (df = 2; 425)	0.004 (df = 2; 515)	2.011 (df = 1; 11)	0.591 (df = 2; 16)

Note:

*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 7 are listed below.

```

1 # Table 7 & 8 Data
2 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")
3 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")
4 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")
5 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")
6 # manually construct DID and THREEINTERACTION
7 data_0m_t7_exp$DID <- data_0m_t7_exp$CCStorePresent * data_0m_t7_exp$AfterStoreClosing
8 data_0m_t7_exp$THREEINTER <- data_0m_t7_exp$CCStorePresent * data_0m_t7_exp$AfterStoreClosing * data_0m_t7_exp$BBStorePresent
9 data_0m_t8_search$DID <- data_0m_t8_search$CCStorePresent * data_0m_t8_search$AfterStoreClosing
10 data_0m_t8_search$THREEINTER <- data_0m_t8_search$CCStorePresent * data_0m_t8_search$AfterStoreClosing * data_0m_t8_search$BBStorePresent
11 data_5m_t7_exp$DID <- data_5m_t7_exp$CCStorePresent * data_5m_t7_exp$AfterStoreClosing

```

```

12 data_5m_t7_exp$THREEINTER <- data_5m_t7_exp$CCStorePresent * data_5m_t7_exp$AfterStoreClosing * data_5m_t7_exp$BBStorePresent
13 data_5m_t8_search$DID <- data_5m_t8_search$CCStorePresent * data_5m_t8_search$AfterStoreClosing
14 data_5m_t8_search$THREEINTER <- data_5m_t8_search$CCStorePresent * data_5m_t8_search$AfterStoreClosing * data_5m_t8_search$BBStorePresent
15 # Table 7
16 ama.t7.pagesperdollar.0mile.exp <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "amazon.com",
], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
17 ama.t7.pagesperdollar.5mile.exp <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t7_exp[data_5m_t7_exp$domain_name == "amazon.com",
], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
18 bb.t7.pagesperdollar.0mile.exp <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "bestbuy.com",
], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
19 bb.t7.pagesperdollar.5mile.exp <- plm(log(PagesPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t7_exp[data_5m_t7_exp$domain_name == "bestbuy.com",
], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
20 ama.t7.minsperdollar.0mile.exp <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "amazon.com",
], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
21 ama.t7.minsperdollar.5mile.exp <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_5m_t7_exp[data_5m_t7_exp$domain_name == "amazon.com",
], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
22 bb.t7.minsperdollar.0mile.exp <- plm(log(MinsPerDollar + 1) ~ DID + THREEINTER, data = data_0m_t7_exp[data_0m_t7_exp$domain_name == "bestbuy.com",
], index = c("Zip_Code", "MonthYear"), model = "within", effect = "twoways")
23 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-5 Miles	Amazon-0 Mile	Amazon-5 Miles	BestBuy-5 Miles
	(1)	(2)	(3)	(4)	(5)	(6)
β_1	0.051 (0.554)	0.181 (0.160)	1.038 (0.752)	0.395 (0.555)	0.151 (0.159)	1.522** (0.711)
β_2	-0.320 (0.713)	-0.214 (0.203)		-0.778 (0.715)	-0.182 (0.203)	
Observations	3,741	4,229	342	3,741	4,229	342
R ²	0.0002	0.001	0.042	0.001	0.001	0.096
Adjusted R ²	-1.439	-1.344	-6.593	-1.438	-1.345	-6.166
F Statistic	0.182 (df = 2; 1533)	0.704 (df = 2; 1802)	1.907 (df = 1; 43)	0.613 (df = 2; 1533)	0.501 (df = 2; 1802)	4.583** (df = 1; 43)

Note:

*p<0.1; **p<0.05; ***p<0.01

Codes for generating Table 8 are listed below.

```

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

```



```

9 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**2.10 Table 10****2.11 Table 11****2.12 Table 12****2.13 Table 13****2.14 Table 14****2.15 Table C1**

Table 9: Change in Demographics after Circuit City Store Closure

Group	Before Store Closure			After Store Closure			First Difference of Mean (p-value)		
	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean
	Age	Income	Education	Age	Income	Education	Age	Income	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)

2.16 Table D1-D4

2.17 Table E1-E2

2.18 Table G1-G3

3 Advanced Method

3.1 Synthetic Control Method

3.2 Generalized Synthetic Control Method

3.3 Heckit Approach

3.4 PSM and LA-PSM

3.5 Measurement Error Bias Correction

4 References

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