## **MIS7420**

# 

Prof. Zhiqiang Zheng, Prof. Vijay Mookerjee

Author: Yihong Liu,

**NETID: YXL180111,** 

## Contents

Li	st of	Figures	3
Li	st of	Tables	4
1	Data	a Cleaning Process	5
2	Pap	er Replication	7
	2.1	Table 1	7
	2.2	Table 2	7
	2.3	Table 3	8
	2.4	Table 4	11
	2.5	Table 5	12
	2.6	Table 6	14
	2.7	Table 7	14
	2.8	Table 8	14
	2.9	Table 9	14
	2.10	Table 10	14
	2.11	Table 11	14
	2.12	Table 12	14
	2.13	Table 13	14
	2.14	Table 14	14
	2.15	Table C1	14
	2.16	Table D1-D4	14
	2.17	Table E1-E2	14
	2.18	Table G1-G3	14
9	م ما م	report Method	1.4
3		ranced Method	14
	3.1	Synthetic Control Method	14
	3.2	Generalized Synthetic Control Method	14
	3.3	Heckit Approach	14
	3.4	PSM and LA-PSM	14
	3.5	Measurement Error Bias Correction	14
4	Refe	erences	15

## List of Figures

## List of Tables

1	Summary Statistics of Top Five Vendors by Sales Volume	7
2	Summary Statistics of Referring Domain Categories	7
3	Average Difference-in-Difference (DID) of the Outcome Variables	8
4	Results of the Sales Effect (All Product Categories)	11
5	Results of the Search Effect (Pages Per Dollar, All Product Categories)	12
6	Results of the Search Effect (Minutes Per Dollar, All Product Categories)	13
7	Change in Demographics after Circuit City Store Closure	13

### 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')
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_Omile_path <- 'data/sales_ccityOmilezipcode_sample.sas7bdat
13 sales_cc_5miles_path <- 'data/sales_ccity5milezipcode_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_Omile <- read_sas(sales_cc_Omile_path)</pre>
19 sales_cc_5miles <- read_sas(sales_cc_5miles_path)
21 # Data Mapping
22 sales_allother_zipcode$Store_Close_Status <- 0 # NaN means no CC in 5-miles radius, we change NaN to 0
24 # Exclude Data without purchase
25 # All data should be with purchase -> tran_flg == 1
26 sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$tran_flg == 1,]
27 sales_cc_Omile <- sales_cc_Omile[sales_cc_Omile$tran_flg == 1,]
28 sales_cc_5miles <- sales_cc_5miles[sales_cc_5miles$tran_flg == 1,]
29
30 # Filter Referring Domain
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 groupby_ref_domain_result <- groupby_ref_domain_result[order(-groupby_ref_domain_result$machine_id),]
35 # we identify some search engines
36 search_engine_to_consider1 <- c("GOOGLE.COM", "YAHOO.COM", "google.com", "yahoo.com",
                                "MSN.COM", "msn.com", "aol.com", "AOL.COM", "LIVE.COM", "live.com",
                                "MYWEBSEARCH.COM". "ASK.COM". "MYWAY.COM". "mywebsearch.com".
38
39
                                "ask.com", "YAHOO.NET", "BIZRATE.COM", "bizrate.com",
                                "amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com",
40
41
                                "AMAZON.COM", "STAPLES.COM", "DELL.COM", "WALMART.COM", "BESTBUY.COM")
43 search_engine_to_consider2 <- c("GOOGLE.COM", "YAHOO.COM", "BING.COM", "google.com", "yahoo.com", "bing.com")
44
45
   ref_domain_to_consider1 <- c("", "GOOGLE.COM", "YAHOO.COM", "google.com", "yahoo.com",
46
                                "MSN.COM", "msn.com", "aol.com", "AOL.COM", "LIVE.COM", "live.com",
                                "MYWEBSEARCH.COM", "ASK.COM", "MYWAY.COM", "mywebsearch.com",
47
48
                                "ask.com", "YAHOO.NET", "BIZRATE.COM", "bizrate.com",
49
                                "amazon.com", "staples.com", "dell.com", "walmart.com", "bestbuy.com",
                                "AMAZON.COM", "STAPLES.COM", "DELL.COM", "WALMART.COM", "BESTBUY.COM")
51
   ref_domain_to_consider2 <- c("", "GOOGLE.COM", "YAHOO.COM", "BING.COM", "google.com", "yahoo.com", "bing.com")
53
54\, # Then we filter data by refer domain name
55 sales_allother_zipcode <- sales_allother_zipcode[(sales_allother_zipcode$ref_domain_name %in% ref_domain_to_consider1),]
56 sales cc Omile <- sales cc Omile ((sales cc Omile ref domain name %in% ref domain to consider1).]
57 sales_cc_5miles <- sales_cc_5miles[(sales_cc_5miles$ref_domain_name %in% ref_domain_to_consider1),]
60 groupby_target_domain_result <- aggregate(machine_id ~ domain_name, rbind(sales_allother_zipcode, sales_cc_5miles), FUN = "length")
61 groupby_target_domain_result <- groupby_target_domain_result[order(-groupby_target_domain_result$machine_id), ]
62 five_target_domain_to_consider <- c("amazon.com", "staples.com", "dell.com", "valmart.com", "bestbuy.com")
63 two_target_domain_to_consider <- c("amazon.com", "bestbuy.com")
65\, # we can choose what filter to apply
66 sales_allother_zipcode <- sales_allother_zipcode[sales_allother_zipcode$domain_name %in% five_target_domain_to_consider,]
67 sales_cc_Omile <- sales_cc_Omile[sales_cc_Omile$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)
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$event_date)
    sales_cc_Omile$event_date <- as.Date(sales_cc_Omile$event_date)
86 sales_cc_5miles$event_date <- as.Date(sales_cc_5miles$event_date)
87
88 # construct MonthYear - month of year
89 sales allother zipcode Month Year <- format (sales allother zipcode Sevent date, "%Y-%m")
90 sales_cc_Omile$MonthYear <- format(sales_cc_Omile$event_date, "%Y-%m")
91
    sales_cc_5miles$MonthYear <- format(sales_cc_5miles$event_date, "%Y-%m")
92
94
95 # CCStorePresent
96 # it is the same as Store_Close_Status
97 \\ \hspace*{0.2cm} \texttt{sales\_allother\_zipcode\$CCStorePresent} \leftarrow \\ \hspace*{0.2cm} \texttt{sales\_allother\_zipcode\$Store\_Close\_Status} \\ \\
    sales_cc_Omile$CCStorePresent <- sales_cc_Omile$Store_Close_Status
99 sales_cc_5miles$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<math>$MonthYear < "2008-11", 0, 1)
104 sales_cc_5miles $AfterStoreClosing <- ifelse(sales_cc_5miles $MonthYear < "2008-11", 0, 1)
105
106 # BBStorePresent
107 sales_allother_zipcode <- merge(sales_allother_zipcode, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)
108 sales_cc_Omile <- merge(sales_cc_Omile, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)
109 sales_cc_5miles <- merge(sales_cc_5miles, bb_zipcode, by.x ="Zip_Code", by.y = "Zip_Code", all.x = TRUE)
110
111 sales_allother_zipcode$BBStorePresent <- na.fill(sales_allother_zipcode$BB_Store_Status, 0)
112 sales cc Omile$BBStorePresent <- na.fill(sales cc Omile$BB Store Status. 0)
113 sales_cc_5miles$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$ref_domain_name == "", 1, 0)
    sales_cc_Omile$NoReferringDomain <- ifelse(sales_cc_Omile$ref_domain_name == "", 1, 0)
119 sales cc 5miles NoReferringDomain <- ifelse (sales cc 5miles ref domain name == "", 1, 0)
120
121 sales_allother_zipcode$ReferringDomainIsSearchEngine <- ifelse(sales_allother_zipcode$ref_domain_name %in% search_engine_to_consider1, 1, 0)
122 sales_cc_Omile$ReferringDomainIsSearchEngine <- ifelse(sales_cc_Omile$ref_domain_name %in% search_engine_to_consider1, 1, 0)
123 sales_cc_5miles$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, ]
```

### 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	${\bf Total Pages Viewed}$	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_Omile_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")
```

#### 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.

```
# Table 2

table2_raw <- rbind(read_sas(sales_allother_zipcode_path), read_sas(sales_cc_Omile_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"</pre>
```

#### 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	Before Store	First Difference	DID
Outcome variable	Groups	Closure	Closure	(se)	
Amazon	Control	3.418	3.303	0.115	
Sales	Control	0.410	5.505	(0.031)	-0.167
Sales	Treatment	3.351	3.403	-0.052	
	Heatment	5.551	5.405	(0.212)	
Amazon	Control	l 1.188 1.147	1 147	0.041	
PagesPerDollar	Control		1.147	(0.025)	0.257
r agesr er Donar	Treatment	1.363	1.065	0.298	
	пеаннен	ment 1.303	1.005	(0.153)	
Amazon	Control	1.016	0.975	0.041	
Amazon MinsPerDollar	Control	1.010	0.975	(0.025)	0.263
MinsPerDonar	Treatment	1.187	0.882	0.304	
	reatment	1.107	0.882	(0.137)	
bestbuy.com	Control	9 /10	2 202	0.354	
Sales	Collitor	Control 3.418 3.303	(0.031)	0.623	
Sales	Treatment	3.351	3.403	0.976	
	reatment			(0.212)	
h oothuu oom	Control	1 100	1.147	-0.109	
bestbuy.com	Control	1.188	1.147	(0.025)	0.074
PagesPerDollar	Treatment	1.363	1.065	-0.035	
	reatment	1.303	1.000	(0.153)	
h	Control	1.016	0.975	-0.084	
bestbuy.com MinsPerDollar	Control	1.016	0.975	(0.025)	-0.012
minsperdonar	Treatment	1.187	0.000	-0.096	
	reatment	1.187	0.882	(0.137)	

Codes for generating Table 3 are listed below.

```
1  # Table 3
2  temp <- read_sas(sales_allother_zipcode_path)
3  temp$Store_Close_Status <- 0
4  table3_Om_raw <- rbind(temp, read_sas(sales_cc_Omile_path))
5  table3_5m_raw <- rbind(temp, read_sas(sales_cc_5miles_path))
6
7  # Date Transform
8  table3_Om_raw$event_date <- as.Date(table3_Om_raw$event_date)
9  table3_5m_raw$event_date <- as.Date(table3_5m_raw$event_date)
10
11  # construct NonthYear - month of year
12  table3_Om_raw$MonthYear <- format(table3_Om_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</pre>
```

```
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
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)
26 # BBStorePresent
   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)
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)
33 # aggregate data
34
35 table3_Om_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 Om 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
42
     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$domain_name == domain_name_used) & (table3_raw$
43
         AfterStoreClosing == 1),]$TotalMonthlySales
44
45
     amazonsales_control_before <- log(amazonsales_control_before + 1)
     amazonsales_control_after <- log(amazonsales_control_after + 1)
46
47
     # t test
48
     t_test.amazonsales_control <- t.test(amazonsales_control_after, amazonsales_control_before)
49
     amazonsales\_control\_mean\_diff\_se \ \leftarrow \ t\_test.amazonsales\_control\$stderr
50
     t_test.amazonsales_control$p.value
5.1
     amazonsales\_control\_after\_mean \ \leftarrow \ 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),]$TotalMonthlySales
     amazonsales_treatment_after <- table3_raw[(table3_raw$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"]]
     amazonsales_treatment_mean_diff <- t_test.amazonsales_treatment$estimate[["mean of x"]] - t_test.amazonsales_treatment$estimate[["mean of y"]]
68
69
70
71
     amazonsales did <- amazonsales treatment mean diff - amazonsales control mean diff
72
73
     # Amazon PagesPerDollar
74
     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),]$TotalPages / table3_rav[(table3_rav$CCStorePresent == 0) & (table3_rav$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
  81
                 t_test.amazonppd_control <- t.test(amazonppd_control_after, amazonppd_control_before)
                 amazonppd_control_mean_diff_se <- t_test.amazonppd_control$stderr
  82
  83
                 t_test.amazonppd_control$p.value
  84
                 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$estimate[["mean of y"]]
  86
  87
  88
                 # Amazon PagesPerDollar
  89
                 # for treatment
                 amazonppd_treatment_before <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
                           AfterStoreClosing == 0),]$TotalPages / table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$domain_name_used) & (table3_raw$domain_name_us
                            raw$AfterStoreClosing == 0),]$TotalMonthlySales
                 amazonppd_treatment_after <- table3_raw[Ctable3_rawsCCStorePresent == 1) & (table3_rawsdomain_name == domain_name_used) & (table3_rawsdomain_name_used) & (tab
  91
                          AfterStoreClosing == 1),]$TotalPages / table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_
                            raw$AfterStoreClosing == 1),]$TotalMonthlySales
 92
                 amazonppd_treatment_before <- log(amazonppd_treatment_before + 1)
  93
  94
                 amazonppd_treatment_after <- log(amazonppd_treatment_after + 1)
  95
                 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$estimate[["mean of y"]]
102
103
                 # Amazon PagesPerDollar DID
104
                 amazonppd_did <- amazonppd_treatment_mean_diff - amazonppd_control_mean_diff
                 # Amazon MinsPerDollar
106
                 # for control
                 amazonmpd_control_before <- table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
108
                           AfterStoreClosing == 0),]$TotalMins / table3_raw[(table3_raw$CCStorePresent == 0) & (table3_raw$domain_name == domain_name_used) & (table3_raw$domain_name_used) & (table3_raw$domain_used) & (table3_raw$domain_used
                            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),]$TotalMonthlySales
                 amazonmpd_control_before <- log(amazonmpd_control_before + 1)
                 amazonmpd_control_after <- log(amazonmpd_control_after + 1)
112
113
                 # t test
114
                 t_test.amazonmpd_control <- t.test(amazonmpd_control_after, amazonmpd_control_before)
                 amazonmpd_control_mean_diff_se <- t_test.amazonmpd_control$stderr
116
                 t_test.amazonmpd_control$p.value
117
                 amazonmpd\_control\_after\_mean <- t\_test.amazonmpd\_control\$estimate[["mean of x"]]
                 amazonmpd control before mean <- t test.amazonmpd control estimate [["mean of v"]]
118
119
                 amazonmpd_control_mean_diff <- t_test.amazonmpd_control$estimate[["mean of x"]] - t_test.amazonmpd_control$estimate[["mean of y"]]
121
                 # Amazon MinsPerDollar
122
                 # for treatment
                 amazonmpd_treatment_before <- table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$
123
                            AfterStoreClosing == 0),]$TotalMins / table3_raw[(table3_raw$CCStorePresent == 1) & (table3_raw$domain_name == domain_name_used) & (table3_raw$domain_name_used) & (tabl
                            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
                 amazonmpd_treatment_before <- log(amazonmpd_treatment_before + 1)
                 amazonmpd_treatment_after <- log(amazonmpd_treatment_after + 1)
127
128
129
                 t_test.amazonmpd_treatment <- t.test(amazonmpd_treatment_after, amazonmpd_treatment_before)
130
                 amazonmpd\_treatment\_mean\_diff\_se \ \leftarrow \ t\_test.amazonmpd\_treatment\$stderr
131
                 \verb|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"]]
                 amazonmpd_treatment_mean_diff <- t_test.amazonmpd_treatment$estimate[["mean of x"]] - t_test.amazonmpd_treatment$estimate[["mean of y"]]
134
135
```

```
# Amazon MinsPerDollar DID
136
137
      amazonmpd_did <- amazonmpd_treatment_mean_diff - amazonmpd_control_mean_diff
138
139
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),
                   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),
                   c(paste(print_name,"PagesPerDollar"),"Treatment", amazonppd_treatment_after_mean, amazonppd_treatment_before_mean, amazonppd_
143
           {\tt treatment\_mean\_diff}, \ {\tt amazonppd\_treatment\_mean\_diff\_se}, \ {\tt amazonppd\_did}),
                   c(paste(print_name, "MinsPerDollar"), "Control", amazonmpd_control_after_mean, amazonmpd_control_before_mean, amazonmpd_control_mean_
144
           diff, amazonmpd_control_mean_diff_se, amazonmpd_did),
                   c(paste(print_name, "MinsPerDollar"), "Treatment", amazonmpd_treatment_after_mean, amazonmpd_treatment_before_mean, amazonmpd_
145
           treatment_mean_diff, amazonmpd_treatment_mean_diff_se, amazonmpd_did))
146
147 }
148
149 # generate table
amazon_table3 <- table3_gen(table3_0m_aggregate, "amazon.com", "Amazon")
bestbuy_table3 <- table3_gen(table3_0m_aggregate, "bestbuy.com", "bestbuy.com")
152
153
    stargazer(rbind(amazon_table3, bestbuy_table3), align=TRUE, summary = FALSE, rownames = FALSE, title="Summary Statistics of Top Five Vendors by
     Sales Volume")
```

#### 2.4 Table 4

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

	Dependent variable:						
	$\log(\text{TotalMonthlySales} + 1)$						
	(1)	(2)	(3)	(4)			
$eta_1$	0.140	-0.206	0.087	-1.020			
	(0.525)	(0.198)	(1.183)	(0.906)			
$eta_2$	0.100	0.094		1.120			
	(0.669)	(0.245)		(1.235)			
Observations	5,089	5,740	725	831			
$\mathbb{R}^2$	0.0002	0.001	0.0001	0.012			
Adjusted R <sup>2</sup>	-1.293	-1.209	-6.869	-6.593			
F Statistic	0.196 (df = 2; 2219)	0.716 (df = 2; 2596)	0.005 (df = 1; 92)	0.653  (df = 2; 108)			

Note:

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

### 2.5 Table 5

Table 5: Results of the Search Effect (Pages Per Dollar, All Product Categories)

	Dependent variable:						
	$\log(\text{PagesPerDollar} + 1)$						
	(1)	(2)	(3)	(4)			
$eta_1$	-0.075	0.075	0.467	0.397			
	(0.413)	(0.157)	(0.677)	(0.511)			
$eta_2$	-0.262	-0.056		-0.045			
	(0.526)	(0.194)		(0.696)			
Observations	5,089	5,740	725	831			
$\mathbb{R}^2$	0.0005	0.0001	0.005	0.009			
Adjusted R <sup>2</sup>	-1.292	-1.211	-6.829	-6.615			
F Statistic	0.531  (df = 2; 2219)	0.120  (df = 2; 2596)	0.477 (df = 1; 92)	0.499 (df = 2; 108)			

Note:

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

Table 6: Results of the Search Effect (Minutes Per Dollar, All Product Categories)

	Dependent variable:						
	$\log({\rm MinsPerDollar})$						
	(1)	(2)	(3)	(4)			
$eta_1$	0.248	0.049	0.165	1.821			
	(0.700)	(0.275)	(1.570)	(1.211)			
$eta_2$	-0.744	-0.089		-0.459			
	(0.928)	(0.336)		(1.650)			
Observations	4,941	5,565	677	775			
$\mathbb{R}^2$	0.0004	0.00003	0.0001	0.031			
Adjusted $\mathbb{R}^2$	-1.321	-1.237	-7.243	-6.811			
F Statistic	0.389 (df = 2; 2128)	0.035 (df = 2; 2487)	0.011 (df = 1; 82)	1.548 (df = 2; 96)			

Note:

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

Table 7: Change in Demographics after Circuit City Store Closure

	Before Store Closure			After Store Closure			First Difference of Mean		
${\bf Group}$							(p-value)		
	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean
	$\mathbf{Age}$	Income	Education	Age	Income	Education	$\mathbf{Age}$	Income	Education
Control	7.048	7.048 4.479 97.957	07.057	6.937	4.498	97.999	-0.111	0.019	0.042
Control			0.331 4.430	4.430	91.999	(<0.0001)	(0.300)	(0.639)	
Treated	7.68	7.69 4.071 09.629	98.632	6.645	4.739	96.843	-1.035	-0.232	-1.789
Treateu	7.68 4.971	90.032	0.040 4	4.139	90.043	(<0.0001)	(0.029)	(0.004)	

- 2.6 Table 6
- 2.7 Table 7
- 2.8 Table 8
- 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
- 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.