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

List of Tables

1 Data Cleaning Process

```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,]
```

```r

# 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, ]
```

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.

```
# 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")
```

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_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"
```

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.

```
# 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_
```

```
    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")
```

2.4 Table 4

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

	Dependent variable:			
	log(TotalMonthlySales + 1)			
	(1)	(2)	(3)	(4)
β_1	0.140	−0.206	0.087	−1.020
	(0.525)	(0.198)	(1.183)	(0.906)
β_2	0.100	0.094		1.120
	(0.669)	(0.245)		(1.235)
Observations	5,089	5,740	725	831
R^2	0.0002	0.001	0.0001	0.012
Adjusted R^2	-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(PagesPerDollar + 1)			
	(1)	(2)	(3)	(4)
β_1	−0.075	0.075	0.467	0.397
	(0.413)	(0.157)	(0.677)	(0.511)
β_2	−0.262	−0.056		−0.045
	(0.526)	(0.194)		(0.696)
Observations	5,089	5,740	725	831
R^2	0.0005	0.0001	0.005	0.009
Adjusted R^2	-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(MinsPerDollar)			
	(1)	(2)	(3)	(4)
β_1	0.248	0.049	0.165	1.821
	(0.700)	(0.275)	(1.570)	(1.211)
β_2	−0.744	−0.089		−0.459
	(0.928)	(0.336)		(1.650)
Observations	4,941	5,565	677	775
R^2	0.0004	0.00003	0.0001	0.031
Adjusted 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

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)

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.