Causal Effect of Data Capture on Retail Sales Analysis

Michael Brucek, Kevin Grady, Anthony Meyers, Danny Moncada, Jack Quick

May 03, 2020

Contents

Data Loading, Munging & Exploratory Data Analysis

Load Data

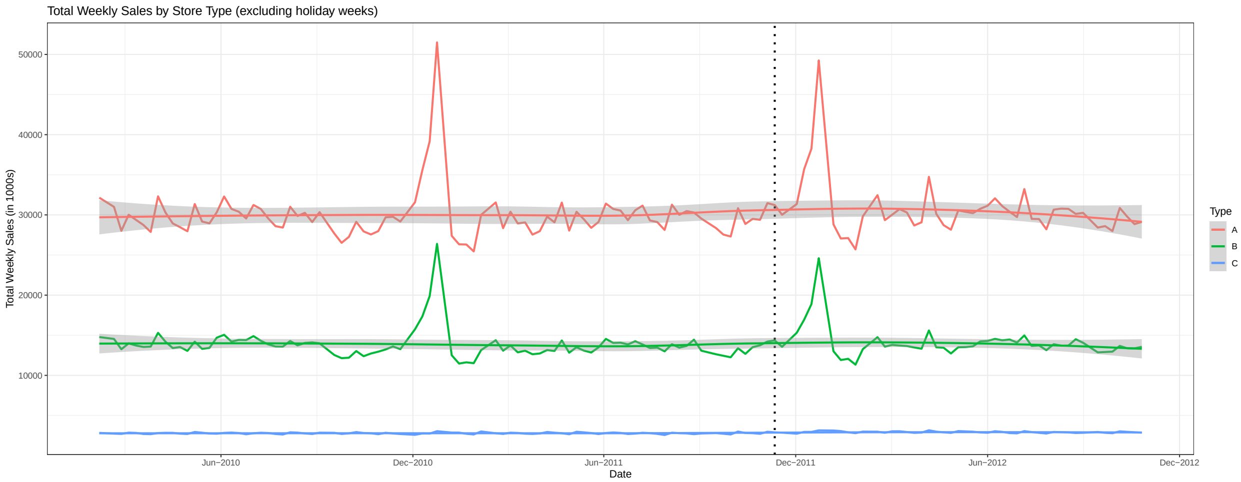
```r
# set the working dir for project files
setwd("C:/Users/danny/Downloads/Causal_Group_Project")

# Original set of data files provided by retailer
stores <- read_csv("stores data-set.csv")
sales <- read_csv("sales data-set.csv")
features <- read_csv("Features data set.csv")

# Data files generated for clustering analysis
stores_agg <- read_csv('store_features.csv')
stores_agg_clusters <- read_csv('store_features_with_cluster.csv')
```

Data Munging & Feature Generation

```r
# Convert Date field in sales from dd/mm/yyyy to yyyy/mm/dd
sales$Date_new <- strptime(as.character(sales$Date), "%d/%m/%Y")
sales$Date <- format(sales$Date_new, "%Y-%m-%d")
sales$Date <- as.Date(sales$Date, format = "%Y-%m-%d")
sales = sales[-c(6)]

# Convert Date field in features from dd/mm/yyyy to yyyy/mm/dd
features$Date_new <- strptime(as.character(features$Date), "%d/%m/%Y")
features$Date <- format(features$Date_new, "%Y-%m-%d")
features$Date <- as.Date(features$Date, format = "%Y-%m-%d")
features = features[-c(13) ]

features <- features %>% filter(!is.na(CPI) | !is.na(Unemployment))

# Get weekly sales for each store
sales_weekly <- sales %>% group_by(Store, Date) %>%
  summarise(sales = sum(Weekly_Sales), holiday = max(IsHoliday))

# Merge sales and stores to get store size and type
sales_weekly <- merge(sales_weekly, stores, by = "Store")

# Merge sales and featurees to get weekly features for each store
sales_weekly <- merge(sales_weekly, features, by = c("Store", "Date"))

# Create an "after" treatment variable
sales_weekly$after <- ifelse(sales_weekly$Date > '2011-11-04', 1, 0)
```

Exploratory Data Analysis

```r
# Aggregate Total Weekly Sales by Date by Store Type into new table sales_weekly_type
sales_weekly_type <- sales_weekly %>% select(Date, sales, Type, holiday) %>%
  group_by(Date, Type, holiday) %>%
  filter(holiday == 0) %>%
  summarise(Average_Weekly_Sales = mean(sales), Total_Weekly_Sales = sum(sales))

# All Store Types - Plot of Total Weekly Sales by Date;
ggplot(sales_weekly_type, aes(x = Date, y = Total_Weekly_Sales/1000, color = Type)) +
  geom_line(size=1) +
  geom_vline(xintercept = as.Date('2011-11-11'), linetype='dotted', size=1) +
  theme_bw() +
  labs(title="Total Weekly Sales by Store Type (excluding holiday weeks)",
       y="Total Weekly Sales (in 1000s)") +
  scale_x_date(breaks = date_breaks("6 months"),
               labels = date_format("%b-%Y")) +
  geom_smooth(method = 'loess')
```



Fixed Effects Regression

```
# Let's try a fixed effect regression.
# Fixed effect for Store
within_reg <- plm(data=sales_weekly, sales ~ as.factor(after) +
                    Temperature + Fuel_Price + CPI + Unemployment,
                  index=c("Store"),effect="individual",model="within")

pooling_reg <- plm(data=sales_weekly, sales ~ as.factor(after) +
                    Temperature + Fuel_Price + CPI + Unemployment,
                  index=c("Store"),effect="individual",model="pooling")

ols_reg <-lm(sales ~ as.factor(after) + Temperature + Fuel_Price + CPI + Unemployment , data = sales_we

# Generate summary table for fixed effects regression
stargazer(within_reg, pooling_reg,
          title="Fixed vs. Pooling Effect Model",
          column.labels = c("Within", "Pooling"), type="text")
```

```
##
## Fixed vs. Pooling Effect Model
## ==================================================================
##                                   Dependent variable:
##                     ----------------------------------------------
##                                          sales
##                              Within                  Pooling
##                               (1)                      (2)
## ------------------------------------------------------------------
## as.factor(after)1        56,749.900***             12,633.090
##                           (8,571.019)             (17,323.690)
##
## Temperature               -825.319***              -855.520**
##                            (133.686)               (398.361)
##
## Fuel_Price              -28,924.700***            -19,037.560
```

```
##                         (7,600.574)            (18,297.890)
##
## CPI                     -3,632.309**          -1,604.655***
##                         (1,481.997)             (196.873)
##
## Unemployment            -9,505.957**          -40,679.720***
##                         (4,713.477)            (4,040.049)
##
## Constant                                     1,759,025.000***
##                                                (82,317.380)
##
## ----------------------------------------------------------------------
## Observations               6,435                    6,435
## R2                         0.028                    0.024
## Adjusted R2                0.020                    0.024
## F Statistic       36.429*** (df = 5; 6385) 32.174*** (df = 5; 6429)
## ======================================================================
## Note:                                   *p<0.1; **p<0.05; ***p<0.01
```

```r
# Let's see if panel data model is needed
pFtest(within_reg, pooling_reg)
```

```
##
##  F test for individual effects
##
## data:  sales ~ as.factor(after) + Temperature + Fuel_Price + CPI + Unemployment
## F = 1618, df1 = 44, df2 = 6385, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

```r
# Significant effects for Store
```

Fixed Effects vs Random Effects

```r
# Fixed Effect vs Random Effect
random_reg = plm(sales ~ as.factor(after) + Temperature +
                   Fuel_Price + CPI + Unemployment,
                 data = sales_weekly, index=c("Store"), effect="individual", model="random")

# Generate summary table comparing fixed effect vs random effect
stargazer(within_reg, random_reg,
          title="Fixed vs. Random Effect Model",
          column.labels = c("Within", "Random"), type="text")
```

```
##
## Fixed vs. Random Effect Model
## ===========================================================
##                              Dependent variable:
##                    ----------------------------------------
##                                    sales
##                            Within              Random
##                             (1)                 (2)
```

```
## -----------------------------------------------------------
## as.factor(after)1      56,749.900***       53,204.400***
##                        (8,571.019)         (7,778.890)
##
## Temperature             -825.319***         -836.927***
##                          (133.686)           (133.169)
##
## Fuel_Price             -28,924.700***      -31,700.070***
##                        (7,600.574)         (7,030.376)
##
## CPI                     -3,632.309**        -2,844.772**
##                        (1,481.997)         (1,215.637)
##
## Unemployment            -9,505.957**        -9,952.908**
##                        (4,713.477)         (4,688.161)
##
## Constant                                  1,752,945.000***
##                                            (215,857.200)
##
## -----------------------------------------------------------
## Observations                6,435               6,435
## R2                          0.028               0.028
## Adjusted R2                 0.020               0.027
## F Statistic       36.429*** (df = 5; 6385)    181.933***
## ===========================================================
## Note:                           *p<0.1; **p<0.05; ***p<0.01
```

```r
# Hausman test
phtest(within_reg, random_reg)
```

```
##
##  Hausman Test
##
## data:  sales ~ as.factor(after) + Temperature + Fuel_Price + CPI + Unemployment
## chisq = 1.2965, df = 5, p-value = 0.9353
## alternative hypothesis: one model is inconsistent
```

```r
# Durbin Wu Hausman prefers random effect model

# Serial Correlation (Breusch Godfrey Test)

# Null hypothesis: Random Effect model is more efficient
# With p-value effectively = 0, we reject and conclude that there is serial correlation
pbgtest(random_reg)
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel
##  models
##
## data:  sales ~ as.factor(after) + Temperature + Fuel_Price + CPI + Unemployment
## chisq = 4041.6, df = 143, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

```r
# Testing for Hetroskedasticity
bptest(sales ~ as.factor(after) + Temperature +
         Fuel_Price + CPI + Unemployment + as.factor(Store),
       data = sales_weekly)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  sales ~ as.factor(after) + Temperature + Fuel_Price + CPI + Unemployment +     as.factor(Store
## BP = 414.87, df = 49, p-value < 2.2e-16
```

```r
# Heteroskedasticity not present
```

```r
random_reg2 <-  plm(log(sales) ~ as.factor(after) + Temperature +
                      Fuel_Price + CPI + Unemployment,
                    data = sales_weekly, index=c("Store"), effect="individual", model="random")

# Generate summary table
summary(random_reg2)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = log(sales) ~ as.factor(after) + Temperature + Fuel_Price +
##     CPI + Unemployment, data = sales_weekly, effect = "individual",
##     model = "random", index = c("Store"))
##
## Balanced Panel: n = 45, T = 143, N = 6435
##
## Effects:
##                   var std.dev share
## idiosyncratic 0.01454 0.12057 0.039
## individual    0.35928 0.59940 0.961
## theta: 0.9832
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -0.6487031 -0.0662292 -0.0091427  0.0480468  0.7904587
##
## Coefficients:
##                     Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)       14.54514157  0.18515068 78.5584 < 2.2e-16 ***
## as.factor(after)1  0.05682405  0.00608980  9.3310 < 2.2e-16 ***
## Temperature       -0.00026061  0.00010018 -2.6014  0.009285 **
## Fuel_Price        -0.01625624  0.00545725 -2.9788  0.002893 **
## CPI               -0.00434433  0.00099848 -4.3509 1.356e-05 ***
## Unemployment      -0.00596283  0.00353053 -1.6889  0.091232 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    96.131
```

```
## Residual Sum of Squares: 93.45
## R-Squared:      0.027886
## Adj. R-Squared: 0.02713
## Chisq: 184.421 on 5 DF, p-value: < 2.22e-16
```

```r
random_reg3 <- plm(log(sales) ~ as.factor(after) + Temperature +
                     Fuel_Price + CPI,
                   data = sales_weekly, index=c("Store"), effect="individual", model="random")

# Generate summary table
summary(random_reg3)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = log(sales) ~ as.factor(after) + Temperature + Fuel_Price +
##     CPI, data = sales_weekly, effect = "individual", model = "random",
##     index = c("Store"))
##
## Balanced Panel: n = 45, T = 143, N = 6435
##
## Effects:
##                   var std.dev share
## idiosyncratic 0.01454 0.12059  0.04
## individual    0.35289 0.59404  0.96
## theta: 0.983
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -0.6486076 -0.0669013 -0.0093726  0.0481851  0.7925839
##
## Coefficients:
##                      Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)        1.4487e+01  1.8146e-01 79.8333 < 2.2e-16 ***
## as.factor(after)1  6.1120e-02  5.5164e-03 11.0797 < 2.2e-16 ***
## Temperature       -2.4281e-04  9.9634e-05 -2.4370  0.014810 *
## Fuel_Price        -1.3977e-02  5.2794e-03 -2.6474  0.008112 **
## CPI               -4.3404e-03  9.9690e-04 -4.3539 1.337e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    96.142
## Residual Sum of Squares: 93.503
## R-Squared:      0.027447
## Adj. R-Squared: 0.026842
## Chisq: 181.464 on 4 DF, p-value: < 2.22e-16
```

```r
#Testing for Hetroskedasticity
bptest(log(sales) ~ as.factor(after) + Temperature + Fuel_Price + CPI + as.factor(Store), data = sales_
```

```
##
##  studentized Breusch-Pagan test
```

```
##
## data:  log(sales) ~ as.factor(after) + Temperature + Fuel_Price + CPI +     as.factor(Store)
## BP = 584.12, df = 48, p-value < 2.2e-16
```

```r
# Still no heteroskedasticity
```

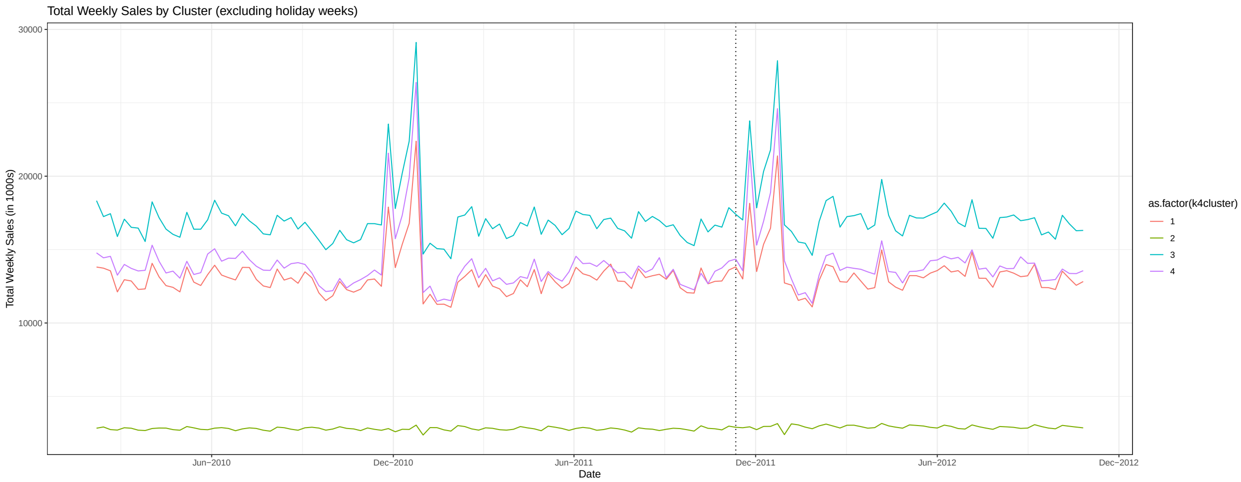
Synthetic Control

Load Data

```r
# Pull in our clustered store data, loaded earlier in the analysis
stores_agg_clusters <- stores_agg_clusters %>% select(Store, k4cluster)

# Merge our two data sets together by store
stores_agg <- merge(stores_agg, stores_agg_clusters, by = 'Store')

# Merge our weekly sales data with our cluster data
sales_weekly_cluster <- merge(sales_weekly, stores_agg_clusters, by = 'Store')

# Create a new dataframe by grouping on our four clusters and date
clusters <- sales_weekly_cluster %>%
  group_by(k4cluster, Date) %>%
  summarise(tot_sales = sum(sales), # Total weekly sales by cluster
            temp = mean(Temperature), # Average temperature by cluster
            fuel_price = mean(Fuel_Price), # Average fuel price by cluster
            CPI = mean(CPI), # Average Consumer Price Index by cluster
            unemployment = mean(unemployment)) # Average unemployment by cluster
```

Visualize Weekly Sales by Cluster

```r
# Generate a plot comparing our clusters and their total sales by week... do they
# behave similarly?
ggplot(clusters) +
  geom_line(aes(Date, tot_sales/1000, col = as.factor(k4cluster))) +
  geom_vline(xintercept = as.Date('2011-11-11'), linetype = 'dotted')+
  theme_bw() +
  labs(title="Total Weekly Sales by Cluster (excluding holiday weeks)",
       y="Total Weekly Sales (in 1000s)") +
  scale_x_date(breaks = date_breaks("6 months"),
               labels = date_format("%b-%Y"))
```



Data Prep

```r
# Read in modeling features data set created for Synthetic control analysis
modeling_features <- read_csv('modeling_features_mb1.csv')

# Subset the CSV file to only include the columns that we need
modeling_features <- modeling_features %>%
              select(week, k4cluster, total_weekly_sales, avg_weekly_sales,
                     total_markdown_count, mean_markdown_count_per_store,
                     total_markdown_sum, mean_markdown_sum_per_store,
                     avg_temp, avg_CPI, avg_unemployment, avg_fuel_price)

# Convert the k4cluster to a numeric value so that it can be used by Synth package
modeling_features$k4cluster <- as.numeric(modeling_features$k4cluster)

# Rename the total_weekly_sales column to Y so that we can actually use this in Synth package
modeling_features$Y <- modeling_features$total_weekly_sales/1000

# Generate a new column with a character field for cluster, again for Synth package
modeling_features$cluster <- ifelse(modeling_features$k4cluster == 1, 'One',
                                    ifelse(modeling_features$k4cluster == 2, 'Two',
                                           ifelse(modeling_features$k4cluster == 3, 'Three', 'Four')))

# Convert to data frame for Synth package
modeling_features <- as.data.frame(modeling_features)

# View a sample of the data table
head(modeling_features[c("week", "cluster", "Y", "avg_temp", "avg_CPI")])
```

```
##   week cluster         Y avg_temp  avg_CPI
## 1    1     One 13812.255 31.32100 129.3674
## 2    1     Two  2831.116 44.61167 167.1660
## 3    1   Three 18331.872 37.53000 204.8923
## 4    1    Four 14775.499 29.43765 164.2654
## 5    2     One 13734.745 31.94200 129.4239
## 6    2     Two  2914.928 43.81000 167.2641
```

Analysis

```
# Outcome variable *must* be named Y for Synth to accept it
dataprep.out=
   dataprep(foo = modeling_features,
   predictors = c("total_markdown_count", "mean_markdown_count_per_store",
                  "total_markdown_sum", "mean_markdown_sum_per_store",
                  "avg_temp","avg_CPI","avg_unemployment","avg_fuel_price"),
   predictors.op = "mean",
   dependent = "Y",
   unit.variable = "k4cluster",
   time.variable = "week",

   # Which panel is treated?
   # Cluster 1
   treatment.identifier = 1,

   # Which panels are we using to construct the synthetic control?
   # The remaining clusters
   controls.identifier = c(2,3,4),

   ## What is the pre-treatment time period?
   ## Weeks 1-93 were before markdown was captured
   time.predictors.prior = c(1:93),

   # The whole time frame of data, from weeks 1 through the data
   time.optimize.ssr = c(1:143),

   # Name of panel units
   unit.names.variable = "cluster",

   # Time period to generate the plot for (all weeks)
   time.plot = c(1:143))

# Output the data
synth.out = synth(dataprep.out)
```

```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 282849.3
##
## solution.v:
##  0.1208459 0.04296734 0.2578876 0.2578876 0.1409373 0.001603996 0.08850074 0.08936961
```

```
##
## solution.w:
##  0.08294942 0.01590458 0.901146
```

Visualization

```r
path.plot(dataprep.res = dataprep.out, synth.res = synth.out,
          Xlab="Week", Ylab="Total Weekly Sales (1000s)",
          Main="Comparison of Synthetic vs. Actual Total Weekly Sales")

abline(v=93,lty=2,col="blue")
```