Supplementary Appendix

Danilo Freire

11 July 2016

This file accompanies the article "Evaluating the Effect of Homicide Prevention Strategies in São Paulo, Brazil: A Synthetic Control Approach" (2016). Please set your working directory to the `data/` folder.

1 Data Wrangling

```
# Clear the workspace
rm(list = ls())

# Load necessary packages
library(reshape2)  # data manipulation

# Dependent variable:
dep <- read.csv("homicide-rates.csv", header = TRUE, skip = 1)

dep.molten <- melt(dep, id.vars = c("Sigla", "Código", "Estado"))

colnames(dep.molten) <- c("abbreviation", "code", "state", "year", "homicide.rates")

dep.molten$year <- as.numeric(substring(dep.molten$year, 2))

# Independent variables
ind1 <- read.csv("state-gdp-capita.csv", header = TRUE, skip = 1)

ind1.molten <- melt(ind1, id.vars = c("Sigla", "Código", "Estado"))

colnames(ind1.molten) <- c("abbreviation", "code", "state", "year", "state.gdp.capita")

ind1.molten$year <- as.numeric(substring(ind1.molten$year, 2))

ind2 <- read.csv("state-gdp-growth-percentage.csv", header = TRUE, skip = 1)

ind2.molten <- melt(ind2, id.vars = c("Sigla", "Código", "Estado"))

colnames(ind2.molten) <- c("abbreviation", "code", "state", "year", "state.gdp.growth.percent")

ind2.molten$year <- as.numeric(substring(ind2.molten$year, 2))

ind3 <- read.csv("gini.csv", header = TRUE, skip = 1)

ind3.molten <- melt(ind3, id.vars = c("Sigla", "Código", "Estado"))

colnames(ind3.molten) <- c("abbreviation", "code", "state", "year", "gini")
```

```r
ind3.molten$year <- as.numeric(substring(ind3.molten$year, 2))

ind4 <- read.csv("population-projection.csv", header = TRUE, skip = 1)

ind4.molten <- melt(ind4, id.vars = c("Sigla", "Código", "Estado"))

colnames(ind4.molten) <- c("abbreviation", "code", "state", "year", "population.projection")

ind4.molten$year <- as.numeric(substring(ind4.molten$year, 2))

ind5 <- read.csv("population-extreme-poverty.csv", header = TRUE, skip = 1)

ind5.molten <- melt(ind5, id.vars = c("Sigla", "Código", "Estado"))

colnames(ind5.molten) <- c("abbreviation", "code", "state", "year", "population.extreme.poverty")

ind5.molten$year <- as.numeric(substring(ind5.molten$year, 2))

ind6 <- read.csv("years-schooling.csv", header = TRUE, skip = 1)

ind6.molten <- melt(ind6, id.vars = c("Sigla", "Código", "Estado"))

colnames(ind6.molten) <- c("abbreviation", "code", "state", "year", "years.schooling")

ind6.molten$year <- as.numeric(substring(ind6.molten$year, 2))

# Merges files
data.list <- list(dep.molten, ind1.molten, ind2.molten, ind3.molten, ind4.molten,
    ind5.molten, ind6.molten)

data1 <- Reduce(function(...) merge(..., all = TRUE), data.list)

# Subset and sort
data2 <- subset(data1, year >= 1990 & year <= 2009)
data2 <- data2[order(data2$state), ]
rownames(data2) <- NULL

# Count missing observations, calculate their percentage
round(sapply(data2, function(x) length(which(is.na(x)))), 2)
```

```
##               abbreviation                       code
##                          0                          0
##                      state                       year
##                          0                          0
##             homicide.rates           state.gdp.capita
##                          0                          0
##   state.gdp.growth.percent                       gini
##                          0                         82
##      population.projection population.extreme.poverty
##                          0                         82
##            years.schooling
##                         82
```

```r
round(sapply(data2, function(x) length(which(is.na(x)))/length(x)), 2)
```

```
##              abbreviation                       code
##                      0.00                       0.00
##                     state                       year
##                      0.00                       0.00
##            homicide.rates           state.gdp.capita
##                      0.00                       0.00
##   state.gdp.growth.percent                       gini
##                      0.00                       0.15
##      population.projection population.extreme.poverty
##                      0.00                       0.15
##            years.schooling
##                      0.15
```

```r
# Linear imputation of missing values.
data2$gini.imp <- approxfun(seq_along(data2$gini), data2$gini)(seq_along(data2$gini))
data2$population.extreme.poverty.imp <- approxfun(seq_along(data2$population.extreme.poverty),
    data2$population.extreme.poverty)(seq_along(data2$population.extreme.poverty))
data2$years.schooling.imp <- approxfun(seq_along(data2$years.schooling), data2$years.schooling)(seq_alon

# Create proportion.extreme.poverty
data2$proportion.extreme.poverty <- data2$population.extreme.poverty.imp/data2$population.projection

# Transform variables to improve interpretation
data2$population.projection.ln <- log(data2$population.projection)

# Save data as df.csv
write.table(data2, "df.csv", row.names = FALSE, col.names = TRUE, sep = ",")
```

2 Data Analysis

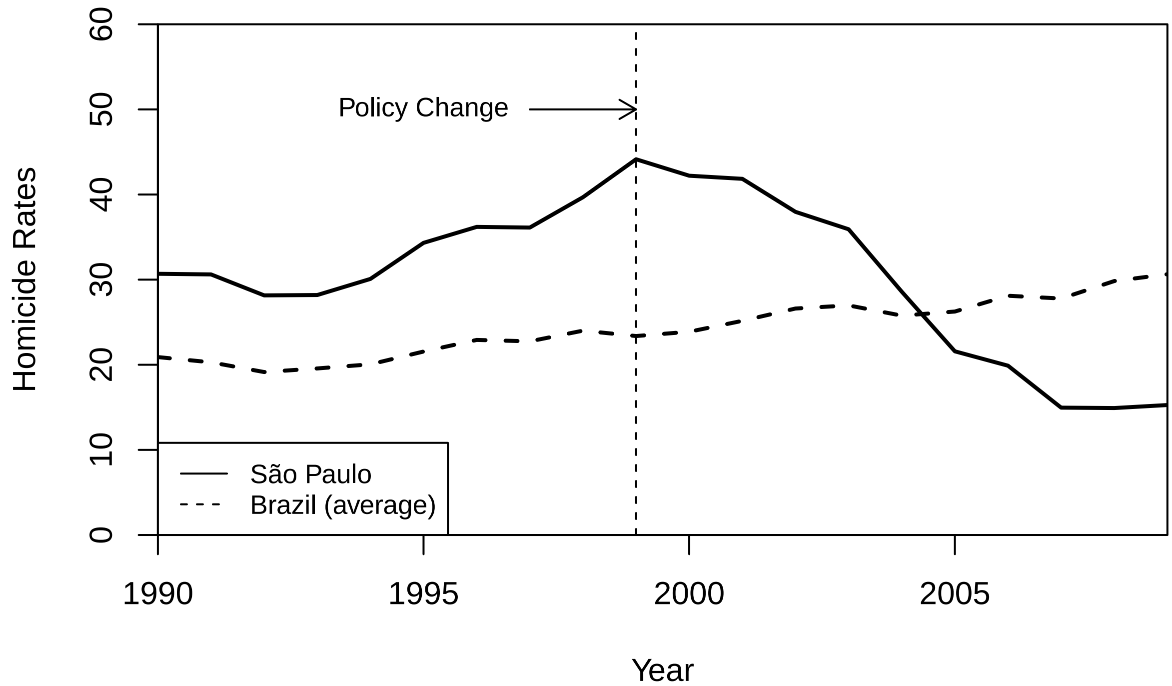
```r
# Load necessary packages
library(dplyr)  # data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(Synth)  # models
```

```
## ##
## ## Synth Package: Implements Synthetic Control Methods.

## ## See http://www.mit.edu/~jhainm/software.htm for additional information.
```

```r
# Load data
df <- read.csv("/home/sussa/Documents/GitHub/homicides-sp-synth/data/df.csv",
    header = TRUE)

# Prepare dataset
df$state <- as.character(df$state)  # required by dataprep()

# Plot: Homicide rates for Sao Paulo and Brazil (average)
df1 <- df %>% mutate(homicide.sp = ifelse(homicide.rates & state == "São Paulo",
    homicide.rates, NA)) %>% select(year, homicide.sp)

df2 <- df %>% mutate(homicide.rates1 = ifelse(homicide.rates & state != "São Paulo",
    homicide.rates, NA)) %>% group_by(year) %>% summarise(homicide.br = mean(homicide.rates1,
    na.rm = TRUE))

plot(x = df1$year, y = df1$homicide.sp, type = "l", ylim = c(0, 60), xlim = c(1990,
    2009), xlab = "Year", ylab = "Homicide Rates", cex = 3, lwd = 2, xaxs = "i",
    yaxs = "i")

lines(df2$year, df2$homicide.br, lty = 2, cex = 3, lwd = 2)

arrows(1997, 50, 1999, 50, col = "black", length = 0.1)

text(1995, 50, "Policy Change", cex = 0.8)

abline(v = 1999, lty = 2)

legend(x = "bottomleft", legend = c("São Paulo", "Brazil (average)"), lty = c("solid",
    "dashed"), cex = 0.8, bg = "white", lwdc(2, 2))
```




```r
# Prepare data for synth
dataprep.out <- dataprep(df, predictors = c("state.gdp.capita", "state.gdp.growth.percent",
    "population.projection.ln", "years.schooling.imp"), special.predictors = list(list("homicide.rates"
    1990:1998, "mean"), list("proportion.extreme.poverty", 1990:1998, "mean"),
    list("gini.imp", 1990:1998, "mean")), predictors.op = "mean", dependent = "homicide.rates",
    unit.variable = "code", time.variable = "year", unit.names.variable = "state",
    treatment.identifier = 35, controls.identifier = c(11:17, 21:27, 31:33,
        41:43, 50:53), time.predictors.prior = c(1990:1998), time.optimize.ssr = c(1990:1998),
    time.plot = c(1990:2009))

# Run synth
synth.out <- synth(dataprep.out)
```

```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 2.660544
##
## solution.v:
##  0.2752884 8.00985e-05 0.0006707994 0.4687482 0.2411453 0.008954685 0.005112477
##
## solution.w:
```

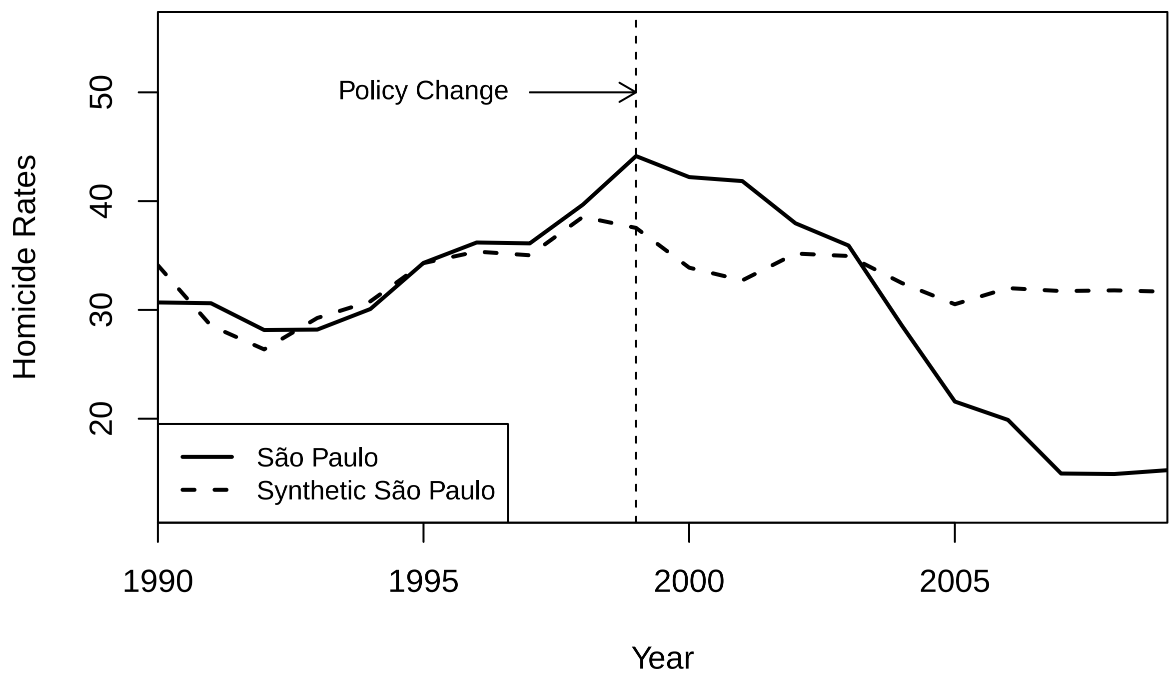
```
##  1.59427e-05 1.04959e-05 1.19579e-05 0.1367322 1.12166e-05 2.60626e-05 3.83051e-05 0.0001724405 4.13
```

```r
# Get result tables
print(synth.tables <- synth.tab(dataprep.res = dataprep.out, synth.res = synth.out))
```

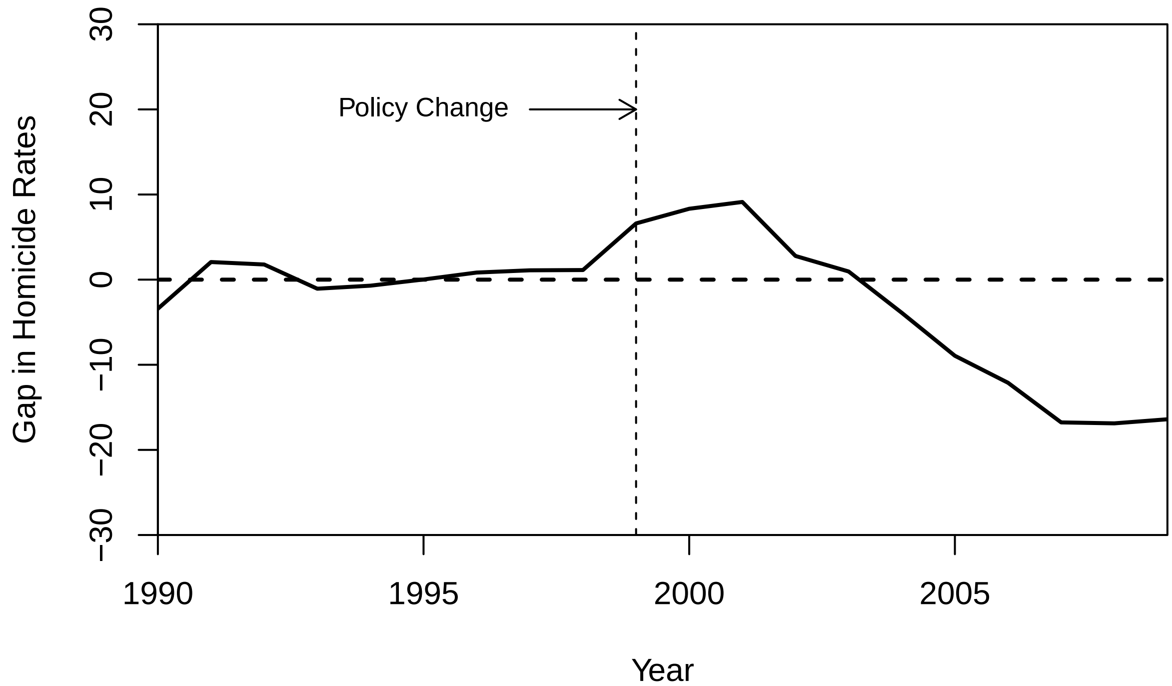
```
## $tab.pred
##                                            Treated Synthetic Sample Mean
## state.gdp.capita                            23.285    23.079      11.830
## state.gdp.growth.percent                     1.330     2.585       3.528
## population.projection.ln                    17.335    14.838      14.867
## years.schooling.imp                          6.089     6.110       4.963
## special.homicide.rates.1990.1998            32.672    32.479      21.843
## special.proportion.extreme.poverty.1990.1998  0.054     0.082       0.185
## special.gini.imp.1990.1998                   0.536     0.561       0.578
##
## $tab.v
##                                            v.weights
## state.gdp.capita                           0.275
## state.gdp.growth.percent                   0
## population.projection.ln                   0.001
## years.schooling.imp                        0.469
## special.homicide.rates.1990.1998           0.241
## special.proportion.extreme.poverty.1990.1998 0.009
## special.gini.imp.1990.1998                 0.005
##
## $tab.w
##    w.weights          unit.names unit.numbers
## 11     0.000            Rondônia           11
## 12     0.000                Acre           12
## 13     0.000            Amazonas           13
## 14     0.137             Roraima           14
## 15     0.000                Pará           15
## 16     0.000               Amapá           16
## 17     0.000           Tocantins           17
## 21     0.000            Maranhão           21
## 22     0.000               Piauí           22
## 23     0.000               Ceará           23
## 24     0.000 Rio Grande do Norte           24
## 25     0.000             Paraíba           25
## 26     0.001          Pernambuco           26
## 27     0.000             Alagoas           27
## 31     0.000        Minas Gerais           31
## 32     0.209      Espírito Santo           32
## 33     0.169      Rio de Janeiro           33
## 41     0.000              Paraná           41
## 42     0.274      Santa Catarina           42
## 43     0.000   Rio Grande do Sul           43
## 50     0.000  Mato Grosso do Sul           50
## 51     0.000         Mato Grosso           51
## 52     0.000               Goiás           52
## 53     0.210    Distrito Federal           53
##
## $tab.loss
##           Loss W   Loss V
```

```
## [1,] 0.007396694 2.660544
```

```r
# Plot: Main model
path.plot(synth.res = synth.out, dataprep.res = dataprep.out, Ylab = c("Homicide Rates"),
    Xlab = c("Year"), Legend = c("São Paulo", "Synthetic São Paulo"), Legend.position = c("bottomleft"))

abline(v = 1999, lty = 2)

arrows(1997, 50, 1999, 50, col = "black", length = 0.1)

text(1995, 50, "Policy Change", cex = 0.8)
```




```r
# Main model: gaps plot
gaps.plot(synth.res = synth.out, dataprep.res = dataprep.out, Ylab = c("Gap in Homicide Rates"),
    Xlab = c("Year"), Ylim = c(-30, 30), Main = "")

abline(v = 1999, lty = 2)

arrows(1997, 20, 1999, 20, col = "black", length = 0.1)

text(1995, 20, "Policy Change", cex = 0.8)
```




```
## Calculating how many lives were saved during the treatment period

# Weights below retrieved form dataprep.out State Code State Weight State
# Name State Abbreviation 42 0.274 Santa Catarina SC 53 0.210 Distrito
# Federal DF 32 0.209 Espirito Santo ES 33 0.169 Rio de Janeiro RJ 14 0.137
# Roraima RR 14 0.001 Pernambuco PB 35 treat Sao Paulo SP

# Get years after policy change
df.2 <- df[which(df$year >= 1999), ]

# Calculate total number of deaths in SP
num.deaths.sp <- sum((df.2$homicide.rates[which(df.2$abbreviation == "SP")])/1e+05 *
    (df.2$population.projection[which(df.2$abbreviation == "SP")]))

# Calculate estimated number of deaths in Synthetic São Paulo
num.deaths.synthetic.sp <- sum((0.274 * (df.2$homicide.rates[which(df.2$abbreviation ==
    "SC")])/1e+05 * (df.2$population.projection[which(df.2$abbreviation == "SP")])) +
    (0.21 * (df.2$homicide.rates[which(df.2$abbreviation == "DF")])/1e+05 *
        (df.2$population.projection[which(df.2$abbreviation == "SP")])) + (0.209 *
    (df.2$homicide.rates[which(df.2$abbreviation == "ES")])/1e+05 * (df.2$population.projection[which(d
    "SP")])) + (0.169 * (df.2$homicide.rates[which(df.2$abbreviation == "RJ")])/1e+05 *
    (df.2$population.projection[which(df.2$abbreviation == "SP")])) + (0.137 *
    (df.2$homicide.rates[which(df.2$abbreviation == "RR")])/1e+05 * (df.2$population.projection[which(d
    "SP")])) + (0.001 * (df.2$homicide.rates[which(df.2$abbreviation == "PB")])/1e+05 *
    (df.2$population.projection[which(df.2$abbreviation == "SP")])))

lives.saved <- num.deaths.synthetic.sp - num.deaths.sp
lives.saved  # Between 1999 and 2009
```

```
## [1] 20331.17
```

3 Robustness Tests

```
## Prepare dataset
df$state <- as.character(df$state) # required by dataprep()

## Placebo Test -- Control ends in 1994
dataprep.out1 <-
        dataprep(df,
                 predictors = c("state.gdp.capita",
                                "state.gdp.growth.percent",
                                "population.projection.ln",
                                "years.schooling.imp"
                 ),
                 special.predictors = list(
                         list("homicide.rates", 1990:1994, "mean"),
                         list("proportion.extreme.poverty", 1990:1994, "mean"),
                         list("gini.imp", 1990:1994, "mean")
                 ),
                 predictors.op = "mean",
                 dependent     = "homicide.rates",
                 unit.variable = "code",
                 time.variable = "year",
                 unit.names.variable   = "state",
                 treatment.identifier  = 35,
                 controls.identifier   = c(11:17, 21:27, 31:33, 41:43, 50:53),
                 time.predictors.prior = c(1990:1994),
                 time.optimize.ssr     = c(1990:1994),
                 time.plot             = c(1990:1998)
                 )

# Run synth
synth.out1 <- synth(dataprep.out1)
```

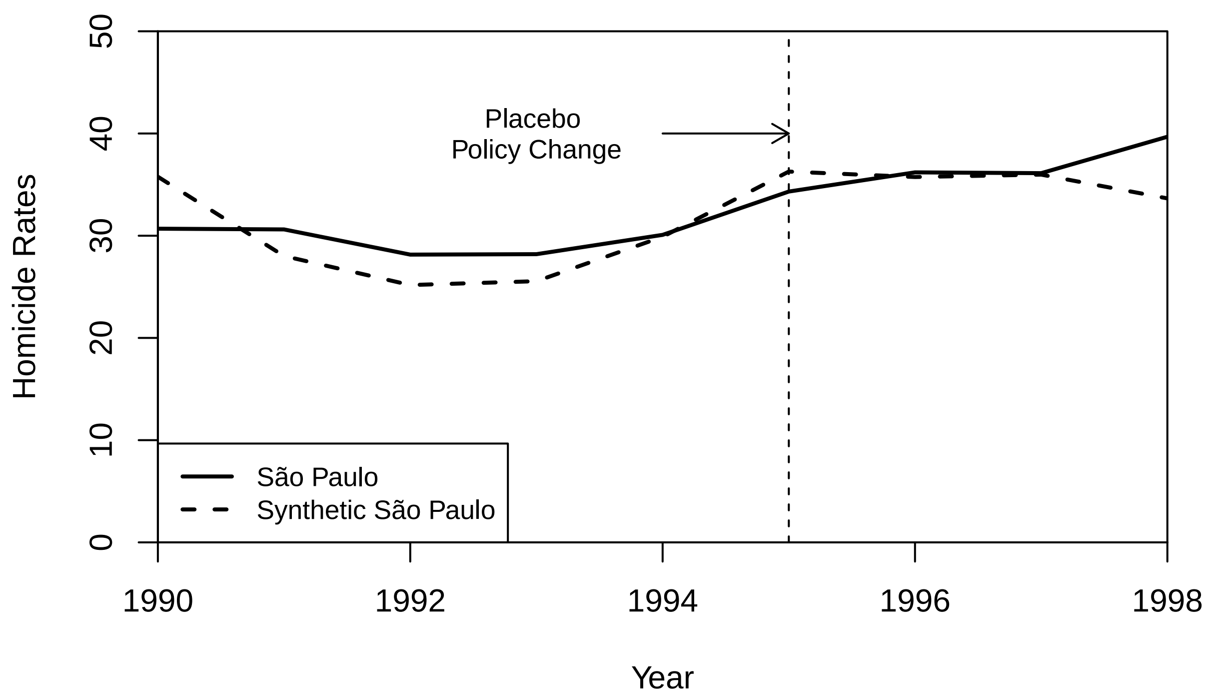
```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 9.696465
##
## solution.v:
##  0.06103391 0.01408394 0.4490766 0.03322913 0.0864128 0.01137627 0.3447874
##
## solution.w:
##  4.4654e-06 1.25e-06 2.94e-06 2.6025e-06 2.9188e-06 2.6755e-06 1.0631e-06 5.2061e-06 4.15e-08 6.178e-
```

```r
# Get result tables
print(synth.tables   <- synth.tab(
        dataprep.res = dataprep.out1,
        synth.res    = synth.out1)
      )
```

```
## $tab.pred
##                                              Treated Synthetic Sample Mean
## state.gdp.capita                              22.569    18.564      11.415
## state.gdp.growth.percent                       0.767     3.189       4.020
## population.projection.ln                      17.301    16.211      14.825
## years.schooling.imp                            5.857     5.954       4.791
## special.homicide.rates.1990.1994              29.546    28.880      20.415
## special.proportion.extreme.poverty.1990.1994   0.062     0.099       0.204
## special.gini.imp.1990.1994                     0.534     0.562       0.577
##
## $tab.v
##                                              v.weights
## state.gdp.capita                             0.061
## state.gdp.growth.percent                     0.014
## population.projection.ln                     0.449
## years.schooling.imp                          0.033
## special.homicide.rates.1990.1994             0.086
## special.proportion.extreme.poverty.1990.1994 0.011
## special.gini.imp.1990.1994                   0.345
##
## $tab.w
##    w.weights          unit.names unit.numbers
## 11     0.000            Rondônia           11
## 12     0.000                Acre           12
## 13     0.000            Amazonas           13
## 14     0.000             Roraima           14
## 15     0.000                Pará           15
## 16     0.000               Amapá           16
## 17     0.000           Tocantins           17
## 21     0.000            Maranhão           21
## 22     0.000               Piauí           22
## 23     0.000               Ceará           23
## 24     0.000 Rio Grande do Norte           24
## 25     0.000             Paraíba           25
## 26     0.000          Pernambuco           26
## 27     0.000             Alagoas           27
## 31     0.000        Minas Gerais           31
## 32     0.000      Espírito Santo           32
## 33     0.458      Rio de Janeiro           33
## 41     0.000              Paraná           41
## 42     0.000      Santa Catarina           42
## 43     0.542   Rio Grande do Sul           43
## 50     0.000  Mato Grosso do Sul           50
## 51     0.000         Mato Grosso           51
## 52     0.000               Goiás           52
## 53     0.000    Distrito Federal           53
##
```

```
## $tab.loss
##           Loss W   Loss V
## [1,] 0.7036748 9.696465
```

```
# Placebo test: graph
path.plot(synth.res       = synth.out1,
          dataprep.res    = dataprep.out1,
          Ylab            = c("Homicide Rates"),
          Xlab            = c("Year"),
          Legend          = c("São Paulo","Synthetic São Paulo"),
          Legend.position = c("bottomleft"),
          Ylim            = c(0, 50)
)

abline(v   = 1995,
       lty = 2)

arrows(1994, 40, 1995, 40,
       col    = "black",
       length = .1)

text(1993, 40,
     "Placebo \nPolicy Change",
     cex = .8)
```




```
## Leave-one-out

# Loop over leave one outs
storegaps <- matrix(NA, length(1990:2009), 4)

colnames(storegaps) <- c(14, 33, 42, 53) # RR, RJ, SC, DF
```

```r
co <- unique(df$code)
co <- co[-25]

for(k in 1:4){

        # Data prep for training model
        omit <- c(14, 33, 42, 53)[k]

        # Prepare data for synth
        dataprep.out2 <-
                dataprep(df,
                         predictors = c("state.gdp.capita",
                                        "state.gdp.growth.percent",
                                        "population.projection.ln",
                                        "years.schooling.imp"
                         ),
                         special.predictors = list(
                                list("homicide.rates", 1990:1998, "mean"),
                                list("proportion.extreme.poverty", 1990:1998, "mean"),
                                list("gini.imp", 1990:1998, "mean")
                         ),
                         predictors.op = "mean",
                         dependent     = "homicide.rates",
                         unit.variable = "code",
                         time.variable = "year",
                         unit.names.variable   = "state",
                         treatment.identifier  = 35,
                         controls.identifier   = co[-which(co==omit)],
                         time.predictors.prior = c(1990:1998),
                         time.optimize.ssr     = c(1990:1998),
                         time.plot             = c(1990:2009)
                )

        # Run synth
        synth.out2 <- synth(dataprep.out2)

        storegaps[,k] <- (dataprep.out2$Y0%*%synth.out2$solution.w)
} # Close loop over leave one outs
```

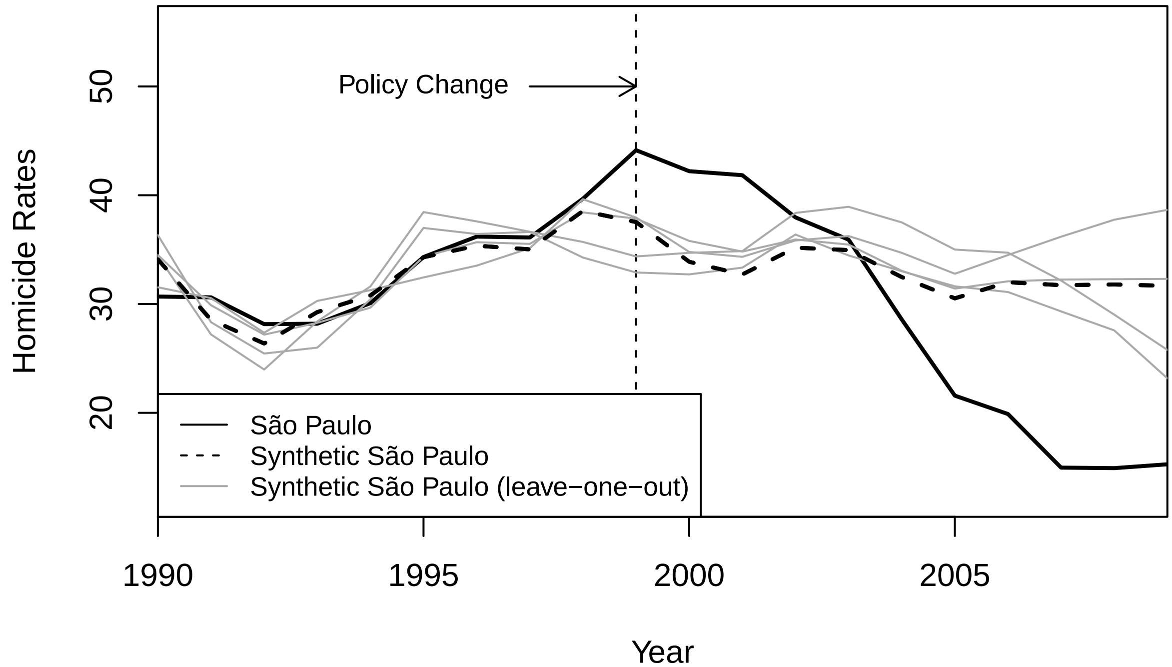
```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 8.94167
##
```

```
## solution.v:
##  0.07950922 0.006980835 0.001923486 0.03109483 0.8575804 0.02290841 2.8345e-06
##
## solution.w:
##  0.000319466 8.88147e-05 0.0002832574 8.04357e-05 0.0001739838 9.28439e-05 0.0001020977 7.83525e-05 (
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 2.086762
##
## solution.v:
##  0.03968127 0.01409698 0.01175056 0.1174728 0.7812639 0.02323605 0.01249846
##
## solution.w:
##  1.98193e-05 1.2506e-06 1e-08 0.0899443 7.10872e-05 5.162e-07 1.2465e-06 4.5909e-06 1.205e-06 4.697e-
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 2.048327
##
## solution.v:
##  0.3008021 0.0001833816 0.001465245 0.1360951 0.5244587 0.03155943 0.005436038
##
## solution.w:
##  4.74279e-05 2.23948e-05 2.36066e-05 0.1248891 3.34217e-05 3.49984e-05 3.60273e-05 6.8763e-05 3.0413
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
```

```
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 9.573976
##
## solution.v:
##  8.3595e-05 8.23008e-05 0.3195644 0.1119996 0.3122513 0.03538738 0.2206315
##
## solution.w:
##  6.41611e-05 1.1159e-05 4.59083e-05 9.51321e-05 7.60632e-05 3.56445e-05 1.5159e-06 9.21467e-05 3.296
```

```
# Leave-one-out: graph
path.plot(synth.res    = synth.out,
          dataprep.res = dataprep.out,
          Ylab         = c("Homicide Rates"),
          Xlab         = c("Year"),
          Legend       = c("São Paulo","Synthetic São Paulo"),
          Legend.position = c("bottomleft")
)

abline(v   = 1999,
       lty = 2)

arrows(1997, 50, 1999, 50,
       col    = "black",
       length = .1)

text(1995, 50,
     "Policy Change",
     cex = .8)

for(i in 1:4){
        lines(1990:2009,
              storegaps[,i],
              col = "darkgrey",
              lty = "solid")
}

lines(1990:2009,
      dataprep.out$Y0plot %*% synth.out$solution.w,
      col = "black",
      lty = "dashed",
      lwd = 2)

legend(x = "bottomleft",
       legend = c("São Paulo",
                  "Synthetic São Paulo",
                  "Synthetic São Paulo (leave-one-out)"
       ),
       lty    = c("solid", "dashed", "solid"),
       col    = c("black", "black", "darkgrey"),
       cex    = .8,
       bg     = "white",
```

```
        lwdc(2, 2, 1)
)
```




```
## Permutation test
states <- c(11:17, 21:27, 31:33, 35, 41:43, 50:53)

# Prepare data for synth
results <- list()
results_synth <- list()
gaps <- list()

for (i in states) {
    dataprep.out <-
            dataprep(df,
                     predictors = c("state.gdp.capita",
                                    "state.gdp.growth.percent",
                                    "population.projection.ln",
                                    "years.schooling.imp"
                                    ),
                     special.predictors = list(
                             list("homicide.rates", 1990:1998, "mean"),
                             list("proportion.extreme.poverty", 1990:1998, "mean"),
                             list("gini.imp", 1990:1998, "mean")
                             ),
                     predictors.op = "mean",
                     dependent     = "homicide.rates",
                     unit.variable = "code",
                     time.variable = "year",
                     unit.names.variable   = "state",
                     treatment.identifier  = i,
                     controls.identifier   = states[which(states!=i)],
                     time.predictors.prior = c(1990:1998),
```

```
                     time.optimize.ssr     = c(1990:1998),
                     time.plot             = c(1990:2009)
                     )
    results[[as.character(i)]] <- dataprep.out
    results_synth[[as.character(i)]] <- synth(results[[as.character(i)]])
    gaps[[as.character(i)]] <- results[[as.character(i)]]$Y1plot - (results[[as.character(i)]]$Y0plot %*

}
```

```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 51.18124
##
## solution.v:
##  0.4802524 0.02982616 0.0001520199 0.004327547 0.2833874 0.2018782 0.0001763244
##
## solution.w:
##  0.328005 8.47e-08 0.5985572 0.0001173175 1.813e-07 4.303e-07 2.8595e-06 5.443e-07 1.868e-07 1.615e-0
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 15.36195
##
## solution.v:
##  0.2072092 0.2717243 0.007653605 0.04019038 0.2556918 0.1474573 0.07007344
##
## solution.w:
##  0.2136222 0.01604535 1.0988e-06 0.4886554 0.2028194 0.000439133 8.57929e-05 0.0002811962 0.00030492
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
```

```
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 1.023702
##
## solution.v:
##  0.0002167872 0.003574624 6.38905e-05 0.03732915 0.8238312 0.122186 0.01279836
##
## solution.w:
##  2.32589e-05 1.49495e-05 6.1847e-06 7.4156e-06 0.07437292 4.1384e-06 2.8846e-06 2.5558e-06 2.48661e-0
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 45.50234
##
## solution.v:
##  0.03941595 4.29e-07 3.1e-08 0.09977601 0.7715451 0.08757297 0.001689552
##
## solution.w:
##  0.6389265 2.9897e-06 6.376e-06 2.71e-07 5.684e-07 2.17e-08 1.4e-08 1.42e-08 1.56e-08 2.8e-08 2.02e-0
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 1.131612
##
## solution.v:
##  0.009058909 0.002165363 0.06489287 0.00049723 0.8456713 0.0002411125 0.07747325
##
## solution.w:
```

```
##  0.09851559 0.02362116 0.01168423 0.02273964 0.01463571 0.006428658 0.41032 0.00395627 0.004055055 0
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 125.909
##
## solution.v:
##  0.008429297 0.07801542 0.03411915 0.00829024 0.3397522 0.001070161 0.5303236
##
## solution.w:
##  7.99847e-05 0.2356338 0.2783961 0.4441709 9.154e-06 8.65e-08 7.2363e-06 7.7221e-06 0.0001163221 0.0
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 2.531519
##
## solution.v:
##  0.2018689 0.0001003565 0.02685221 0.2053734 0.2448247 0.1378164 0.183164
##
## solution.w:
##  0.0002101146 0.1301066 6.3085e-06 0.002382446 7.53521e-05 0.07289869 0.0007211795 0.7030899 0.00012
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
```

```
## MSPE (LOSS V): 1.941073
##
## solution.v:
##  0.1354017 0.0002290707 0.05259146 0.02408425 0.7196521 0.001853635 0.06618776
##
## solution.w:
##  2.51511e-05 1.5002e-06 1.1564e-06 9.88475e-05 0.2831712 9.0198e-06 2.7168e-06 0.6349326 4.96056e-05
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 10.52683
##
## solution.v:
##  0.05920883 5.5966e-06 2.4329e-06 0.3957179 0.2712242 0.273824 1.70533e-05
##
## solution.w:
##  1.15e-08 3.01e-08 5.98e-08 1.01e-08 2.49e-08 2.32e-08 3.803e-07 0.9999973 2.641e-07 2.809e-07 1.39e
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 1.395551
##
## solution.v:
##  0.02054103 0.0004488899 0.002656136 0.1155899 0.8409767 0.001947175 0.01784012
##
## solution.w:
##  3.8532e-06 5.6022e-06 5.6931e-06 2.6755e-06 6.2028e-06 4.1485e-06 1.02051e-05 1.296e-07 0.6272635 5
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
```

```
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.3111036
##
## solution.v:
##  0.06707774 2.2729e-06 0.06024271 0.04332524 0.793015 0.01681319 0.01952386
##
## solution.w:
##  0.001114646 0.1584875 0.01832858 0.001050629 0.06920098 0.001678484 0.007784012 0.001028427 0.588509
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 2.398635
##
## solution.v:
##  0.02403733 0.0799591 0.02985437 0.01666788 0.8359928 1.92642e-05 0.01346926
##
## solution.w:
##  1.42492e-05 0.0001736899 0.0001691743 4.9468e-06 0.0004627081 1.80822e-05 2.23079e-05 2.5e-09 0.6950
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 12.24763
##
## solution.v:
##  0.01300257 0.003995686 0.0002318422 0.0009931233 0.9284519 0.00026095 0.05306393
##
## solution.w:
##  0.00100095 0.0003491557 0.00039892 0.0005495408 0.000209798 0.0006441076 0.0001311558 0.0001750023 8
##
```

```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 16.46456
##
## solution.v:
##  1.482e-06 0.5247996 0.001868916 5.71e-07 0.197269 4.0755e-06 0.2760564
##
## solution.w:
##  0.0008948425 0.000814387 0.0004240316 0.0007324103 0.0006471134 0.000679961 0.000574746 0.000518628
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.1330523
##
## solution.v:
##  0.006679297 0.007276181 0.0004525123 0.0002093622 0.961381 0.001653868 0.02234777
##
## solution.w:
##  3.6765e-06 8.1859e-06 1.23281e-05 2.9086e-06 0.1191527 5.3241e-06 1.58998e-05 1.16216e-05 0.4882883
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 9.26792
##
```

```
## solution.v:
##  0.1189319 5.2e-08 0.00121216 0.5307538 0.3491021 2.6e-09 2e-09
##
## solution.w:
##  0.0002644413 0.0001517431 0.0001032955 0.0004354566 0.0001110638 0.0002519273 0.0001794025 0.000155
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 136.2062
##
## solution.v:
##  0.009817057 0.004778745 0.0003457267 0.0255394 0.6874439 0.1341365 0.1379386
##
## solution.w:
##  2.76e-08 1.15e-08 9.2e-09 0.1742107 6.6e-09 2.07e-08 3.2e-09 3e-09 1e-09 2.6e-09 2.9e-09 3.2e-09 0
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 2.660544
##
## solution.v:
##  0.2752884 8.00985e-05 0.0006707994 0.4687482 0.2411453 0.008954685 0.005112477
##
## solution.w:
##  1.59427e-05 1.04959e-05 1.19579e-05 0.1367322 1.12166e-05 2.60626e-05 3.83051e-05 0.0001724405 4.13
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
```

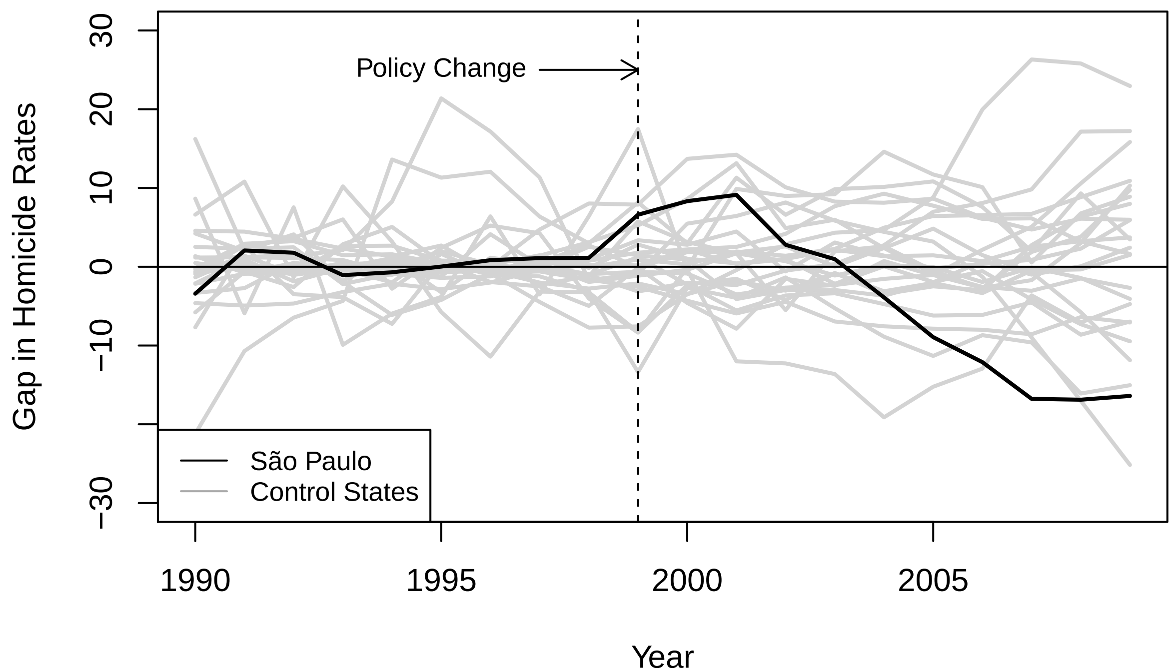
```
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.3488299
##
## solution.v:
##  0.08338114 0.0006088288 0.0001804534 0.05949748 0.8303003 0.01068098 0.01535078
##
## solution.w:
##  0.01543545 0.01575345 0.01985569 0.01301287 0.0203155 0.01475386 0.01317075 0.01067765 0.00708076 0
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.9376258
##
## solution.v:
##  0.003989675 0.05620025 0.2531837 0.002999281 0.6811902 0.002434872 2.0246e-06
##
## solution.w:
##  0.0001184755 4.0719e-06 1.0061e-06 0.0001194974 0.0001964575 0.0001634784 0.3051045 0.0001288999 0.
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 3.411059
##
## solution.v:
##  0.01934869 0.003624292 1.50324e-05 0.04852164 0.8946242 0.02206592 0.01180027
##
## solution.w:
##  0.01161567 2.35942e-05 0.272808 0.008479335 0.1428922 0.002565765 0.0003299683 0.0002213289 9.84241
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
```

```
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 10.01539
##
## solution.v:
##  0.04584823 0.01071171 0.03697681 0.07269026 0.8016851 0.0004406679 0.03164723
##
## solution.w:
##  0.03562505 3.9453e-05 0.007830964 5.78583e-05 0.02230194 0.2322016 0.04059846 0.02249796 0.01647109
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 19.43253
##
## solution.v:
##  0.0749929 0.009337386 0.2437826 0.1916299 0.251775 0.07123169 0.1572506
##
## solution.w:
##  4.3083e-05 7.055e-07 0.03000746 0.1328822 4.24775e-05 7.24e-07 7.058e-07 0.000468338 1.0843e-06 0.2
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 6.51759
##
## solution.v:
##  0.05430354 0.0479132 0.04788227 1e-10 0.7892736 0.000126689 0.06050068
```

```
##
## solution.w:
##  0.01939807 0.03178305 0.01536983 0.01585099 0.3718213 0.01885982 0.001561923 6.92549e-05 0.02215141
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 2.924211
##
## solution.v:
##  6.4036e-06 0.006280579 0.2943406 0.02569878 0.2991516 0.2620339 0.1124882
##
## solution.w:
##  2e-10 0.5030373 2.6795e-06 0 2.7151e-06 4e-10 4.106e-07 7.9e-08 2.345e-07 5.478e-07 7.128e-07 1.739
```

```
# Permutation test: graph
## Permutation test
plot(1990:2009,
     ylim = c(-30, 30),
     xlim = c(1990,2009),
     ylab = "Gap in Homicide Rates",
     xlab = "Year"
)

for (i in states) {
        lines(1990:2009,
              gaps[[as.character(i)]],
              col = "lightgrey",
              lty = "solid",
              lwd = 2
        )
}

lines(1990:2009,
      gaps[["35"]], # São Paulo
      col = "black",
      lty = "solid",
      lwd = 2
)

abline(v   = 1999,
       lty = 2)

abline(h   = 0,
       lty = 1,
```

```
       lwd = 1)

arrows(1997, 25, 1999, 25,
       col    = "black",
       length = .1)

text(1995, 25,
     "Policy Change",
     cex = .8)

legend(x = "bottomleft",
       legend = c("São Paulo",
                  "Control States"),
       lty  = c("solid", "solid"),
       col  = c("black", "darkgrey"),
       cex  = .8,
       bg   = "white",
       lwdc(2, 2, 1)
)
```

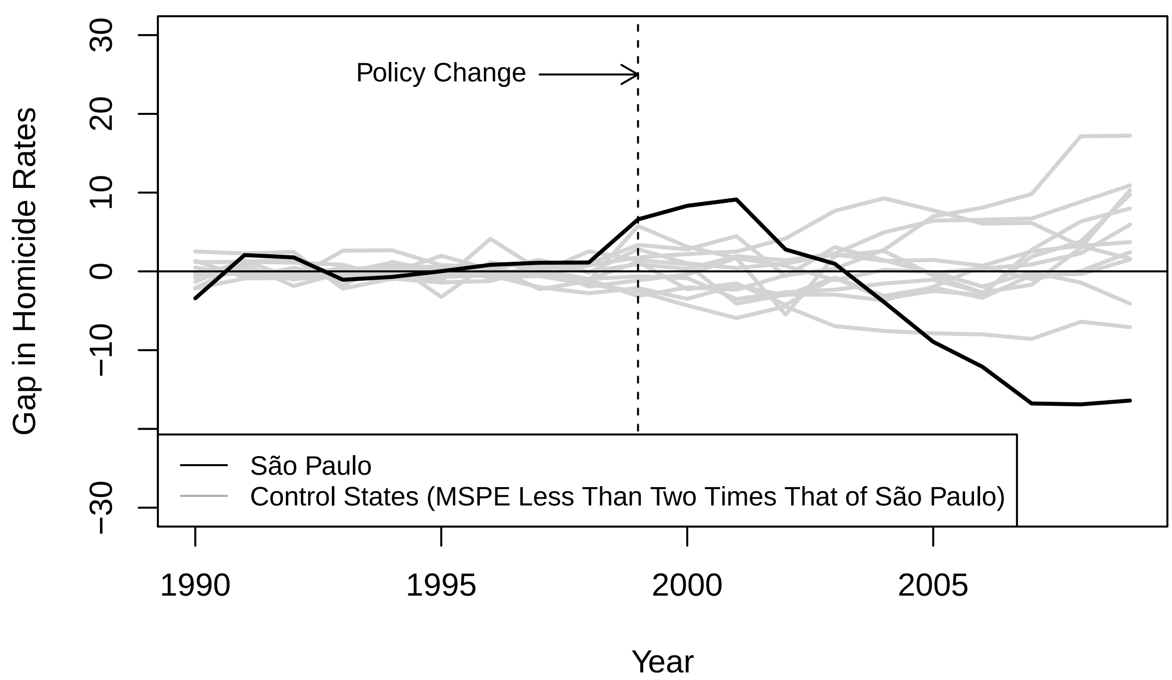



```
# Permutation graph: states with MSPE no higher than 2x São Paulo's
low.mspe <- c(13, 15, 17, 21, 23, 24, 25, 31, 41:43, 53)

plot(1990:2009,
     ylim = c(-30, 30),
     xlim = c(1990,2009),
     ylab = "Gap in Homicide Rates",
     xlab = "Year"
)

for (i in low.mspe) {
lines(1990:2009,
```

```r
      gaps[[as.character(i)]],
      col = "lightgrey",
      lty = "solid",
      lwd = 2
      )
}

lines(1990:2009,
      gaps[["35"]], # São Paulo
      col = "black",
      lty = "solid",
      lwd = 2
      )

abline(v   = 1999,
       lty = 2)

abline(h   = 0,
       lty = 1,
       lwd = 1)

arrows(1997, 25, 1999, 25,
       col    = "black",
       length = .1)

text(1995, 25,
     "Policy Change",
     cex = .8)

legend(x = "bottomleft",
       legend = c("São Paulo",
                  "Control States (MSPE Less Than Two Times That of São Paulo)"),
       lty    = c("solid", "solid"),
       col    = c("black", "darkgrey"),
       cex    = .8,
       bg     = "white",
       lwdc(2, 2, 1)
)

## CausalImpact
# Uncomment the lines below to install the necessary packages
# install.packages(c("devtools", "dtw"))
# library(devtools)
# install_github("google/CausalImpact")
# install_github("klarsen1/MarketMatching", build_vignettes=TRUE)

# Load packages
library(CausalImpact)
```

```
## Loading required package: bsts

## Loading required package: BoomSpikeSlab

## Loading required package: Boom
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: xts

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last
```



```r
library(MarketMatching)
library(ggplot2)

# Prepare data
df$year2 <- as.Date(paste(df$year, sep = "", "-01-01"))

# Estimate model
mm <- best_matches(data=df,
                   id_variable="code",
                   date_variable="year2",
                   matching_variable="homicide.rates",
                   parallel=TRUE,
                   warping_limit=1, # warping limit=1
                   dtw_emphasis=1, # rely only on dtw for pre-screening
                   matches=5, # request 5 matches
                   start_match_period="1990-01-01",
                   end_match_period="1998-01-01")

# View best matches
subset(mm$BestMatches, code == 35) # SP
```

```
##     code BestControl RelativeDistance Correlation Length rank
## 96    35          53        0.1101811  0.62764509      9    1
## 97    35          50        0.2117732  0.81836068      9    2
## 98    35          14        0.2951220  0.25443735      9    3
## 99    35          16        0.3090787  0.69484948      9    4
## 100   35          27        0.3485658 -0.07103921      9    5
##     MatchingStartDate MatchingEndDate
## 96         1990-01-01      1998-01-01
## 97         1990-01-01      1998-01-01
## 98         1990-01-01      1998-01-01
## 99         1990-01-01      1998-01-01
## 100        1990-01-01      1998-01-01
```

```r
# Results
results <- MarketMatching::inference(matched_markets = mm,
                                     test_market = "35",
                                     end_post_period = "2009-01-01")
```

```
##  ------------- Inputs -------------
##  Test Market: 35
##  Control Market 1: 14
##  Control Market 2: 16
##  Control Market 3: 27
##  Control Market 4: 50
##  Control Market 5: 53
##  Market ID: code
##  Date Variable: year2
##  Matching (pre) Period Start Date: 1990-01-01
##  Matching (pre) Period End Date: 1998-01-01
##  Post Period Start Date: 1999-01-01
##  Post Period End Date: 2009-01-01
##  Matching Metric: homicide.rates
```

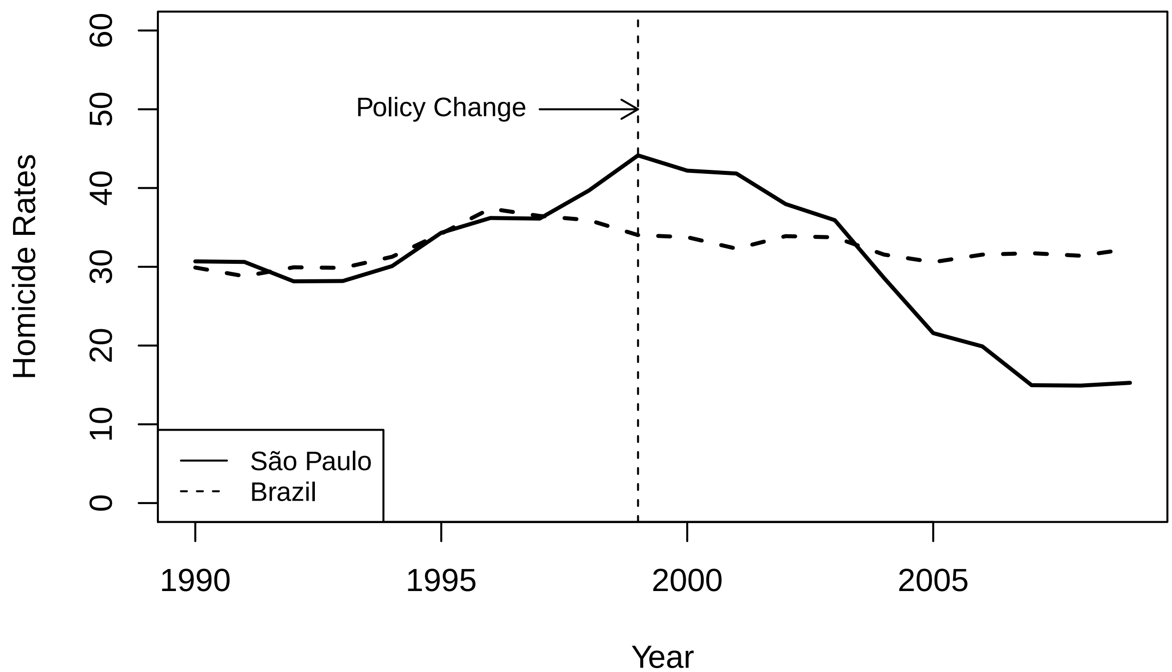
```
##  Local Level Prior SD: 0.01
##  Posterior Intervals Tail Area: 95%
##
##
##  ------------- Model Stats -------------
##  Matching (pre) Period MAPE: 4.43%
##  Beta 1 [14]: 0.199
##  Beta 2 [16]: 0.0959
##  Beta 3 [27]: -0.1499
##  Beta 4 [50]: 0.252
##  Beta 5 [53]: 0.1463
##  DW: 1.79
##
##
##  ------------- Effect Analysis -------------
##  Absolute Effect: -39.54 [-71.4, 14.46]
##  Relative Effect: -11.08% [-20.01%, 4.05%]
##  Probability of a causal impact: 96.3%
```

```
# Predictions
results$Predictions
```

```
##                  Date Response Predicted lower_bound upper_bound
## 1990-01-01 1990-01-01 30.68641  29.90283    24.57575    35.36689
## 1991-01-01 1991-01-01 30.61516  28.80248    24.74202    33.06119
## 1992-01-01 1992-01-01 28.15164  29.93232    25.81587    33.95472
## 1993-01-01 1993-01-01 28.19159  29.86478    25.31562    34.25105
## 1994-01-01 1994-01-01 30.08388  31.26819    27.09969    35.87385
## 1995-01-01 1995-01-01 34.32108  34.21236    30.09417    38.39256
## 1996-01-01 1996-01-01 36.19672  37.38461    33.00364    41.75539
## 1997-01-01 1997-01-01 36.11866  36.45712    31.64439    41.02752
## 1998-01-01 1998-01-01 39.68088  35.92172    31.67007    40.98479
## 1999-01-01 1999-01-01 44.14142  34.02475    28.64683    40.15408
## 2000-01-01 2000-01-01 42.20898  33.75514    29.78423    37.68551
## 2001-01-01 2001-01-01 41.84150  32.30371    27.80812    36.75987
## 2002-01-01 2002-01-01 37.96454  33.89077    29.25718    38.42743
## 2003-01-01 2003-01-01 35.91640  33.73837    28.25152    38.26861
## 2004-01-01 2004-01-01 28.58354  31.54960    25.66870    36.69133
## 2005-01-01 2005-01-01 21.57861  30.59060    24.49848    36.32971
## 2006-01-01 2006-01-01 19.89002  31.54604    23.15682    37.52297
## 2007-01-01 2007-01-01 14.96271  31.72688    21.60589    37.88988
## 2008-01-01 2008-01-01 14.91528  31.39915    19.63272    37.86348
## 2009-01-01 2009-01-01 15.26915  32.28488    22.41630    37.84060
```

```
# Plot results
plot(x = (1990:2009),
     y = as.numeric(results$Predictions$Response),
     type = "l",
     ylim = c(0, 60),
     xlim = c(1990, 2009),
     xlab = "Year",
     ylab = "Homicide Rates",
     cex = 3,
```

```
      lwd = 2)

lines(x = (1990:2009),
      y = as.numeric(results$Predictions$Predicted),
      type = "l",
      lty = 2,
      cex = 3,
      lwd = 2)

arrows(1997, 50, 1999, 50,
       col    = "black",
       length = .1)

text(1995, 50,
     "Policy Change",
     cex = .8)

abline(v   = 1999,
       lty = 2)

legend(x = "bottomleft",
       legend = c("São Paulo",
                  "Brazil"),
       lty    = c("solid", "dashed"),
       cex    = .8,
       bg     = "white",
       lwdc(2, 2)
)
```



4 Session Info

```r
sessionInfo()
```

```
## R version 3.3.0 (2016-05-03)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04 LTS
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggplot2_2.1.0        MarketMatching_0.0.1 CausalImpact_1.0.5
##  [4] bsts_0.6.3           xts_0.9-7            zoo_1.7-13
##  [7] BoomSpikeSlab_0.6.0  Boom_0.4             MASS_7.3-45
## [10] Synth_1.1-5          dplyr_0.5.0          reshape2_1.4.1
##
## loaded via a namespace (and not attached):
##  [1] kernlab_0.9-24       lattice_0.20-33      colorspace_1.2-6
##  [4] htmltools_0.3.5      yaml_2.1.13          chron_2.3-47
##  [7] DBI_0.4-1            optimx_2013.8.7      foreach_1.4.3
## [10] plyr_1.8.4           stringr_1.0.0        munsell_0.4.3
## [13] dfoptim_2011.8-1     gtable_0.2.0         codetools_0.2-14
## [16] setRNG_2013.9-1      evaluate_0.9         Rvmmin_2013-11.12
## [19] knitr_1.13           doParallel_1.0.10    parallel_3.3.0
## [22] Rcpp_0.12.5          scales_0.4.0         formatR_1.4
## [25] BB_2014.10-1         svUnit_0.7-12        digest_0.6.9
## [28] stringi_1.1.1        numDeriv_2014.2-1    dtw_1.18-1
## [31] grid_3.3.0           quadprog_1.5-5       tools_3.3.0
## [34] magrittr_1.5         lazyeval_0.2.0       proxy_0.4-16
## [37] tibble_1.1           ucminf_1.1-3         optextras_2013-10.28
## [40] data.table_1.9.6     Rcgmin_2013-2.21     assertthat_0.1
## [43] minqa_1.2.4          rmarkdown_1.0        iterators_1.0.8
## [46] R6_2.1.2             compiler_3.3.0
```