synth_-_Apr15.R

danny

2020-04-15

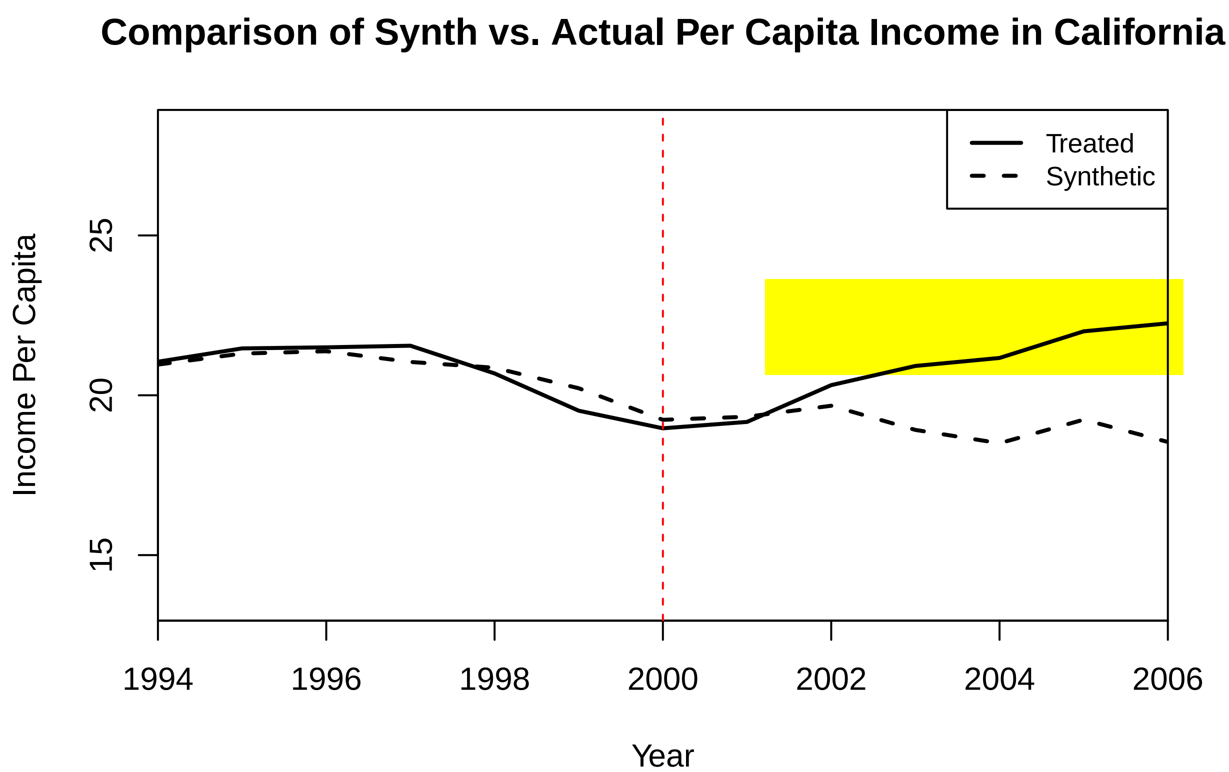
```r
# Authors: Gord Burtch and Gautam Ray
# Course: MSBA 6440 - Causal Inference
# Date: April, 2020
# Topic: Synthetic Control

## install.packages(c("Synth", "ggthemes"))

suppressWarnings(suppressPackageStartupMessages({require(Synth)
require(ggplot2)
require(ggthemes)
}))

#Change the read-in line to wherever your saved version of the fracking data csv file lives
#Note: your panel unit 'names' variable must be a character / string, not a factor, or it won't work.
fracking.data = read.csv("C:/Users/danny/Downloads/fracking.csv",stringsAsFactors=FALSE)
View(fracking.data)

fracking.data$treated = (fracking.data$state=="California")
ggplot(fracking.data, aes(x=year,y=Y,group=state)) +
  geom_line(aes(color=treated,linetype=!(treated))) +
  geom_vline(xintercept=2000,linetype="dashed")
```


```r
#Let's drop the ID column.
fracking.data = fracking.data[,-c(1)]

# your outcome variable *must* be named Y for Synth to accept it (bad coding practices in
# here I suspect)
dataprep.out=
  dataprep(foo = fracking.data,
  predictors = c("res.share", "edu", "pop.dense"),
  predictors.op = "mean",
  dependent = "Y",
  unit.variable = "panel.id",
  time.variable = "year",

  #Any pre-period X's we want to include using different aggregation function, other than
  # mean, or different time windows, specific years vs. all years, we enter here.
  special.predictors = list(list("Y", 1999, "mean"),list("Y", 1995, "mean"),list("Y", 1990, "mean")),

  #which panel is treated?
  treatment.identifier = 7,

  #which panels are we using to construct the synthetic control?
  controls.identifier = c(29, 2, 13, 17, 32, 38),

  #what is the pre-treatment time period?
  time.predictors.prior = c(1994:1999),
```

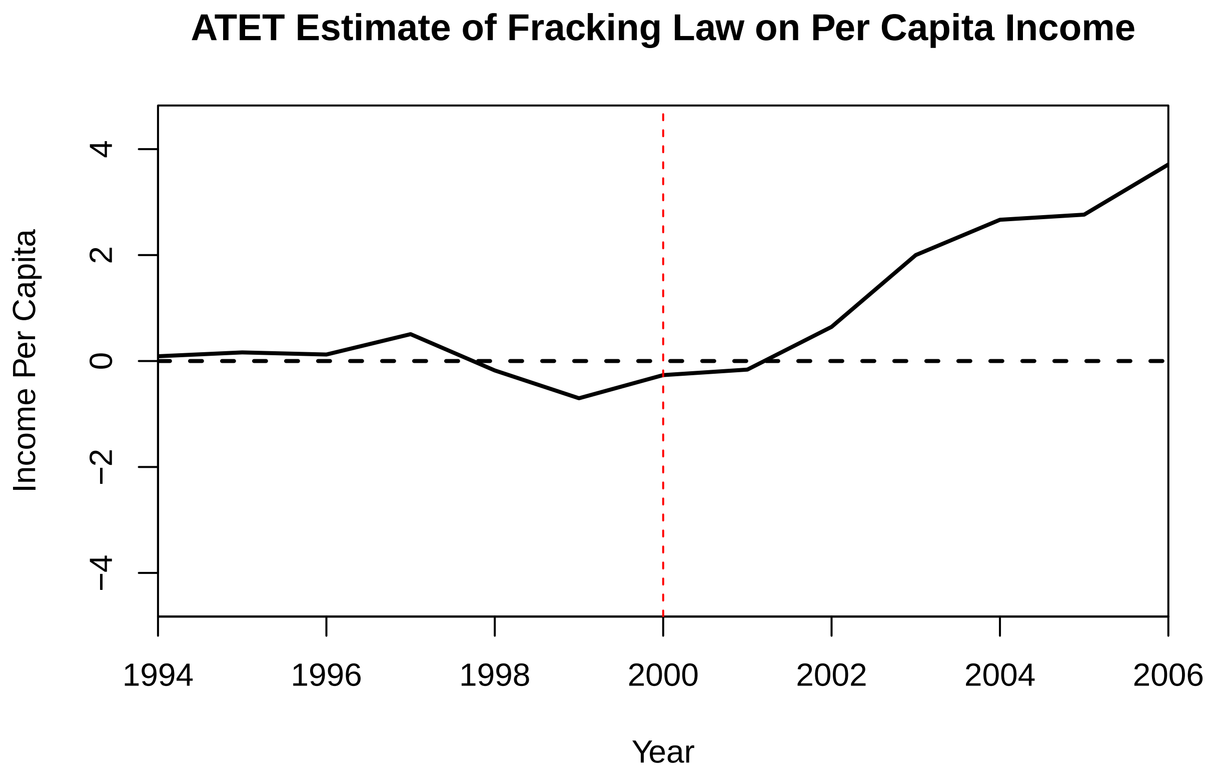
```r
    time.optimize.ssr = c(1994:1999),

    #name of panel units
    unit.names.variable = "state",

    #time period to generate the plot for.
    time.plot = 1994:2006)

synth.out = synth(dataprep.out)
```

```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.1387035
##
## solution.v:
##  2.59612e-05 0.001955033 0.5012642 0.002919795 0.0005281146 0.4933069
##
## solution.w:
##  0.2574452 0.01879814 3.48127e-05 0.1457779 0.4939386 0.08400536
```

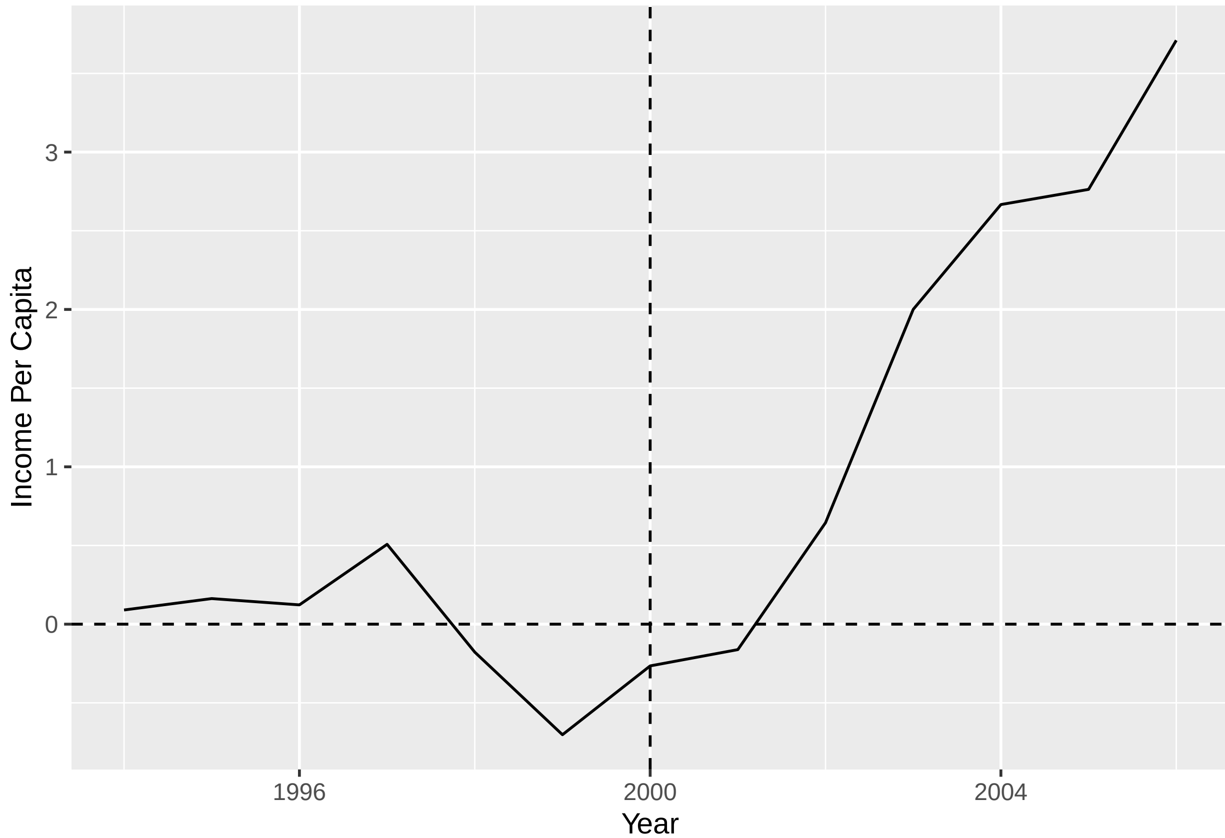
```r
# Two native plotting functions.
# Path.plot() plots the synthetic against the actual treated unit data.
path.plot(dataprep.res = dataprep.out, synth.res = synth.out,Xlab="Year",
          Ylab="Income Per Capita",
          Main="Comparison of Synth vs. Actual Per Capita Income in California")
abline(v=2000,lty=2,col="red")
```




```r
# Gaps.plot() shows the deviation between the synthetic and the actual over time.
gaps.plot(dataprep.res = dataprep.out, synth.res = synth.out,Xlab="Year",
          Ylab="Income Per Capita",Main="ATET Estimate of Fracking Law on Per Capita Income")
abline(v=2000,lty=2,col="red")
```




```r
controls <- c(29, 2, 13, 17, 32, 38)

# We can pull out the data from the result, to make our own nicer plots in ggplot of course
plot.df = data.frame(dataprep.out$Y0plot%*%synth.out$solution.w)
years = as.numeric(row.names(plot.df))
plot.df = data.frame(y=fracking.data$Y[fracking.data$state=='California' &
                              fracking.data$year %in% years]) - data.frame(y=plot.df$w.weight)
plot.df$years <- years
plot.df$state <- "California"
ggplot(plot.df,aes(y=y,x=years)) +
  geom_line() +
  geom_hline(yintercept=0,linetype="dashed") +
  geom_vline(xintercept=2000,linetype="dashed") + xlab("Year") + ylab("Income Per Capita")
```




```
# Okay, let's simulate a null distribution
# We'll run synthetic control on each of the untreated units, using the other units as
# controls (we exclude the treated unit from the control set in each placebo run).
for (i in 1:length(controls)){
  controls_temp <- controls[!controls %in% controls[i]]
  #your outcome variable *must* be named Y for Synth to accept it (bad coding practices in
  # here I suspect)
  dataprep.out.placebo=
    dataprep(foo = fracking.data,
             predictors = c("res.share", "edu", "pop.dense"),
             predictors.op = "mean",
             dependent = "Y",
             unit.variable = "panel.id",
             time.variable = "year",

             #Any pre-period X's we want to include using different aggregation function,
             # other than mean, or different
             # time windows, specific years vs. all years, we enter here.
             special.predictors = list(list("Y", 1999, "mean"),
                                       list("Y", 1995, "mean"),
                                       list("Y", 1990, "mean")),

             # which panel is treated?
             treatment.identifier = controls[i],

             # which panels are we using to construct the synthetic control?
```

```r
             controls.identifier = controls_temp,

             # what is the pre-treatment time period?
             time.predictors.prior = c(1994:1999),

             time.optimize.ssr = c(1994:1999),

             # name of panel units
             unit.names.variable = "state",

             # time period to generate the plot for.
             time.plot = 1994:2006)

  synth.out.placebo = synth(dataprep.out.placebo)
  plot.df.temp <- data.frame(dataprep.out.placebo$Y0plot%*%synth.out.placebo$solution.w)
  years = as.numeric(row.names(plot.df.temp))
  plot.df.update <- data.frame(y=fracking.data$Y[fracking.data$panel.id==controls[i] &
                      fracking.data$year %in% years]) - data.frame(y=plot.df.temp$w.weight)
  plot.df.update$years <- years
  plot.df.update$state <- unique(fracking.data[fracking.data$panel.id==controls[i],]$state)
  plot.df <- rbind(plot.df, plot.df.update)
}
```

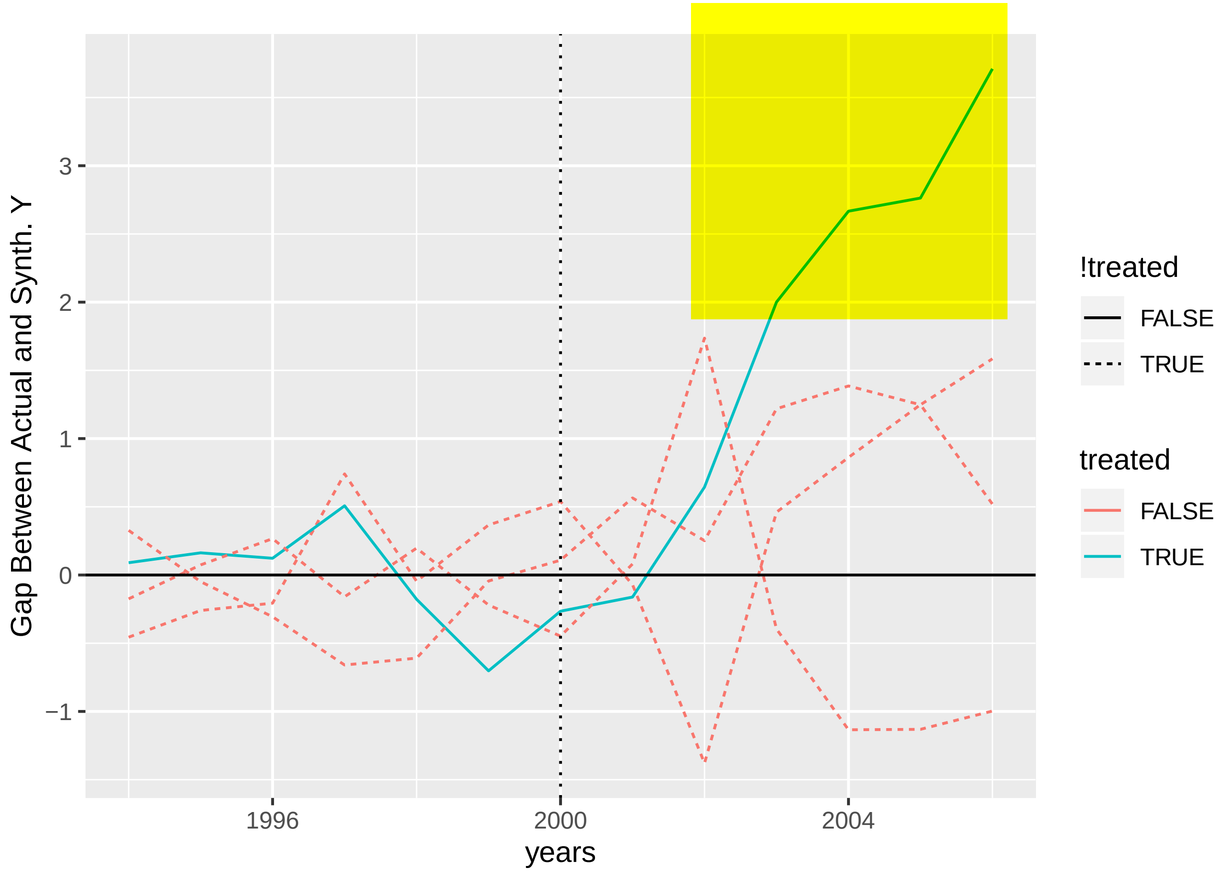
```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.7891398
##
## solution.v:
##  0.1953372 0.1280436 0.5913547 0.007033043 0.07719718 0.001034261
##
## solution.w:
##  5.6635e-06 0.364165 0.1109425 0.0001380115 0.5247488
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
```

```
## ****************
##
## MSPE (LOSS V): 0.03662159
##
## solution.v:
##  0.008192141 0.2159541 0.2063148 0.1875261 0.1993554 0.1826575
##
## solution.w:
##  0.05248441 0.4598678 7.2708e-06 0.4876399 6.843e-07
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 57.62437
##
## solution.v:
##  0.01105524 0.03814458 0.01982471 0.2023291 0.311199 0.4174474
##
## solution.w:
##  5.242e-07 9.46e-08 0.9999988 5.234e-07 2.4e-08
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.1685874
##
## solution.v:
##  0.0136038 0.003942 1.7308e-06 0.5238024 0.4373643 0.02128577
##
## solution.w:
##  0.01719944 0.002975143 0.1419645 0.1980392 0.6398217
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
```

```
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 0.1673348
##
## solution.v:
##  0.0007177705 0.006915393 0.001917603 0.6964686 0.04423186 0.2497487
##
## solution.w:
##  0.9211801 0.07879777 1.796e-07 1.21924e-05 9.7362e-06
##
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 1.700606
##
## solution.v:
##  0.0003706307 0.004005661 0.03501926 0.2383374 0.3876867 0.3345803
##
## solution.w:
##  2.9e-09 4.4e-09 0.9999995 4.315e-07 2.58e-08
```

```r
plot.df$treated <- (plot.df$state=="California")

# Let's plot the diffs associated with each control state.
ggplot(plot.df,aes(y=y,x=years,group=state)) +
  geom_line(aes(color=treated,linetype=!treated)) +
  geom_vline(xintercept=2000,linetype="dotted") +
  geom_hline(yintercept=0)
```


```r
# Our syntheses for ID, OR and IL are all terrible in the pre period.
# I can remove them here for now, but you'd want to tweak the inputs to try to get a better
# MSPE for those three.
ggplot(plot.df[plot.df$state!="Idaho" & plot.df$state!="Oregon" & plot.df$state!="Illinois",],
       aes(y=y,x=years,group=state)) +
  geom_line(aes(color=treated,linetype=!treated)) +
  geom_vline(xintercept=2000,linetype="dotted") +
  geom_hline(yintercept=0) +
  ylab("Gap Between Actual and Synth. Y")
```




```
# I can also recover my cumulative alpha (the ATT) for CA and all placebo estimates.
# by summing over the gaps in the post period.
# If I exclude the 3 poorly synthesized states, CA is the biggest effect in the distribution.
# This is a sparse null distribution, but technically empirical p-value = 0.000.
post.treats <- plot.df[plot.df$year>=2000,]
alphas <- aggregate(post.treats[-c(2:3)], by=list(post.treats$state),FUN=sum)
View(alphas[alphas$Group.1!="Idaho" & alphas$Group.1!="Oregon" & alphas$Group.1!="Illinois",])
```