ARE213 Problem Set #2B

Peter Alstone & Frank Proulx

November 18, 2013

1 Part A: Preliminaries

1.1 I: Comparison between TU and control States

Starting with simple comparisons: We begin with simple comparisons between the dependent outcome of interest, the natural logarithm of traffic fatalities per capita (log(fatalities per capita)), between a predefined composite treatment state "TU" (or, state #99), and all of the potential control states. The mean over the period before primary seatbelt laws were adopted in the treatment state is -1.4 and the mean for the control states is -1.7, indicating approximately a 30% lower typical fatalities rate in the treatment state than the average control state (even before the primary seatbelt law "treatment"). The trends for both shown in Figure 1 show that overall the fatalities were on the decline in both places before the treatment period.

Roadmap: Extracting meaningful conclusions from these data is the goal of our analysis, which will require identifying the variation in traffic fatalities that can be attributed to seat belt laws. Confounding our analysis is the fact that these data are not in the context of an RCT but are from the "real world" with messy trends and linked systems that determine outcomes. We will be applying the synthetic controls method to identify a fleet of control states as a meaningful counterfactual to measure against for our composite treatment state.

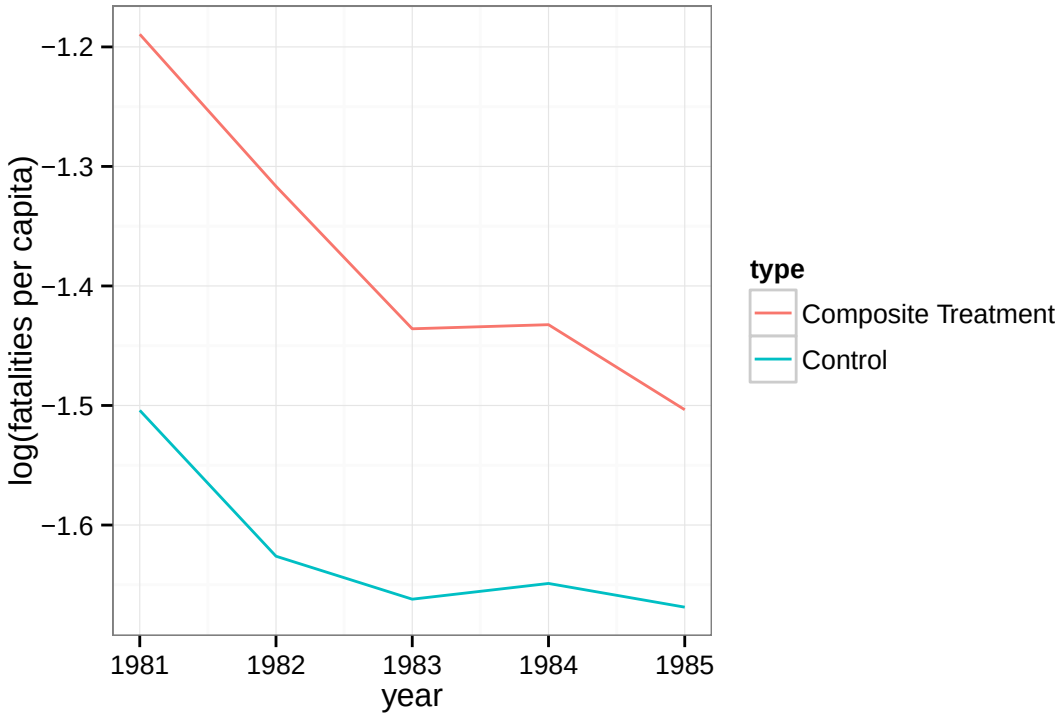

Figure 1: Trend in the dependent variable (log(fatalities per capita)) for the composite treatment state and the average of the control states.

1.2 II: "Best" control state comparison

Sweet Home Alabama: We observe that Alabama is the best match for the composite treatment state based on a simple comparison of log(fatalities per capita) in the year before treatment in the composite state (1985). Figure 2 below shows the distribution in the dependent variable

Fried green covariates and other stereotypes confirmed: Tables 1 and 2 compare the covariates for the composite treatment state and Alabama. There are broad differences between the states. Alabama has higher precipitation, lower college achievement, lower alcohol consumption, higher unemployment, etc. Additionally, the mean value for the depdentent variable of interest, log(fatalities per capita), is quite different for the two states. Examining the trends in the covariates (and dependent variable) for the two states (see Figure 3) shows that it could be construed as a coincidence that Alabama is the best match for the value of the dependent variable, since the trajectory in fatalities for both states are following opposite trends in that time and 1985 happens to be the time when they intersect. There are also important and long-term differences in precipitation and alcohol consumption.

Overall Alabama does not appear to be a particularly good match for the composite treatment state, motivating an application of synthetic controls methods to produce a better match.

2 Part B: Synthetic Controls

2.1 II: Why synthetic controls?

Unsweet Home Alabama: We saw earlier the difficulties in selecting an exact counterfactual match for implementing differences in differences type selection on unobservables techniques. While Alabama would appear on face value to be a good match (based on having similar outcomes in the year prior to treatment) we saw that this was coincidental and that the covariates are not a good match to the composite treatment state. Synthetic control methods are motivated by producing a "better" match by combining (synthesizing) multiple control states in a weighting scheme to create a composite control state with better match of the important covariates and dependent variable than any particular control state.

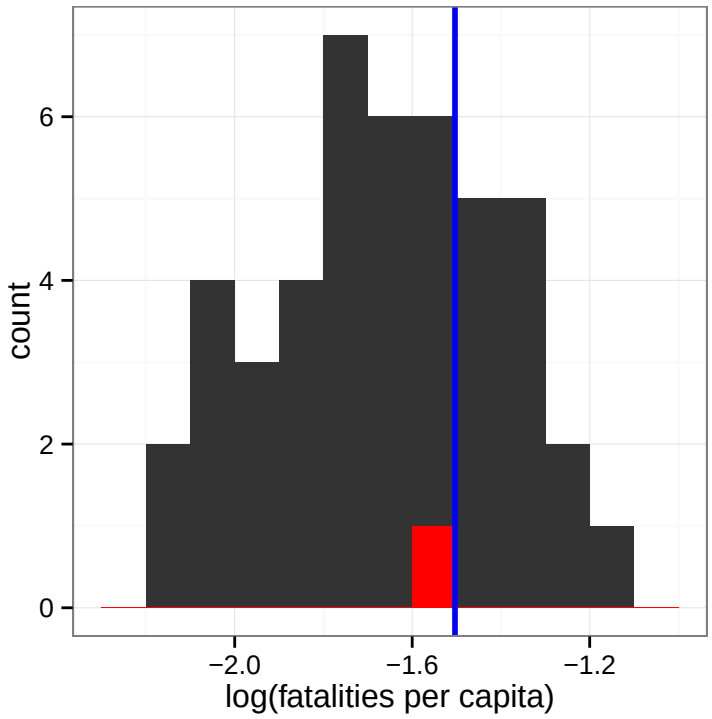

Figure 2: Distribution in traffic fatalities metric from 1985 for all control states with a vertical blue line indicating the value of the metric for the composite treatment state. The red block highlights the position of Alabama in the distribution. Alabama is the closest match to the composite treatment state for 1985, but as is shown here is one of about 11 states that is within 10% of the target value.

Table 1: Composite Treatment Group Summary

Statistic	N	Mean	St. Dev.	Min	Max
state	23	99.000	0.000	99	99
year	23	1,992.000	6.782	1,981	2,003
college	23	0.234	0.014	0.209	0.259
beer	23	1.507	0.074	1.394	1.670
primary	23	0.783	0.422	0	1
secondary	23	0.000	0.000	0	0
population	23	13,597.660	1,813.520	10,737.810	16,862.220
unemploy	23	6.085	1.124	3.855	8.014
fatalities	23	2,619.014	258.667	2,246.977	3,268.613
totalvmt	23	128,099.600	26,447.260	86,013.140	170,407.300
precip	23	2.502	0.289	1.990	3.104
snow32	23	0.143	0.058	0.013	0.270
rural_speed	23	63.443	6.568	55.000	72.886
urban_speed	23	59.184	5.858	55.000	67.138
logfatalpc	23	−1.643	0.168	−1.805	−1.189
sqyears	23	3,968,108.000	27,020.830	3,924,361	4,012,009

Table 2: Closest match for pre-policy fatalities: Alabama

Statistic	N	Mean	St. Dev.	Min	Max
state	23	1.000	0.000	1	1
year	23	1,992.000	6.782	1,981	2,003
college	23	0.170	0.029	0.131	0.220
beer	23	1.105	0.067	1.000	1.190
primary	23	0.174	0.388	0	1
secondary	23	0.304	0.470	0	1
population	23	4,185.794	209.389	3,918.533	4,501.862
unemploy	23	7.509	2.780	4.200	14.400
fatalities	23	1,036.957	88.042	839	1,189
totalvmt	23	44,826.090	10,109.350	27,852	58,637
precip	23	4.944	0.701	3.737	6.342
snow32	23	0.000	0.000	0	0
rural_speed	23	63.696	6.255	55	70
urban_speed	23	58.478	4.870	55	65
logfatalpc	23	−1.398	0.079	−1.543	−1.286
sqyears	23	3,968,108.000	27,020.830	3,924,361	4,012,009

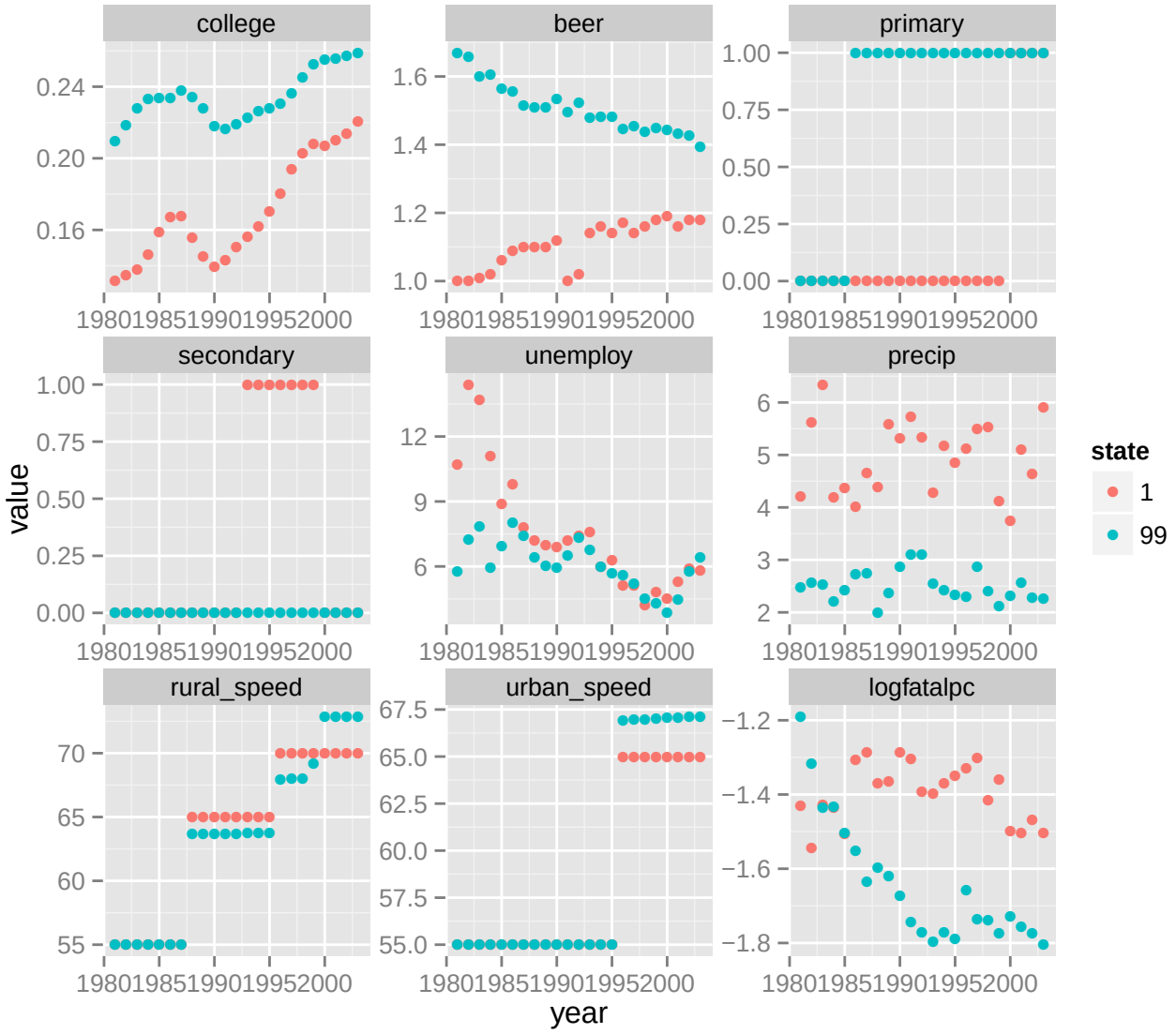

Figure 3: Trends in the covariate (and dependent) variables for the composite treatment state (99) and Alabama (1)

Dr. Synth-love, or how I learned to stop worrying and love econometrics: Synthetic controls have a multi-step, iterative process for developing weighting factors to apply to control states for construction of a composite control state. The goal is to identify a weighting matrix V that minimizes the distance between the treatment covariates with the weighted control unit. The steps are:

1. step1...NOTE: fill out this section :)

Synthesizing control: The process of creating a synthetic control unit involves 2 steps in the Synth package on [R]. First is specifying the form of the model in a "data prep" step. This is then passed to the synthetic control function to attempt implementing the algorithm described above. In practice we found that errors arise when predictors are included that do not have variation in the mean values among the control units. We used an additive process (adding more and more predictor covariates in the specification) to test whether there is variation. A sub-finding is that the computational intensity increases as covariates are added. This is a relatively small dataset but it is possible that this method could become computationally difficult with large datasets and many covariates. After the process of adding we found that there is variation in all the potentially meaningful covariates except rural and urban speed limits. Since speed limits were constant throughout the sample before 1986 they cannot be included in the synthetic controls specification.

3 Appendix: Code Listings

```
1  # Econometrics helper functions for [R]
2  #
3  # Peter Alstone and Frank Proulx
4  # 2013
5  # version 1
6  # contact: peter.alstone AT gmail.com
7
8  # Category: Data Management -------------
9
10
11 # Category: Data Analysis ----------------
12
13 # Function: Find adjusted R^2 for subset of data
14 # This requires a completed linear model...pull out the relevant y-values
       and residuals and feed them to function
```

```
# [TODO @Peter] Improve function so it can simply evaluate lm or glm object, add error handling, general clean up.
adjr2 <- function(y,resid){
  r2 <- 1-sum(resid^2) / sum((y-mean(y))^2)
  return(r2)
} #end adjr2


# Category: Plots and Graphics -------------

## Function for arranging ggplots. use png(); arrange(p1, p2, ncol=1); dev.off() to save.
require(grid)
vp.layout <- function(x, y) viewport(layout.pos.row=x, layout.pos.col=y)
arrange_ggplot2 <- function(..., nrow=NULL, ncol=NULL, as.table=FALSE) {
  dots <- list(...)
  n <- length(dots)
  if(is.null(nrow) & is.null(ncol)) { nrow = floor(n/2) ; ncol = ceiling(n/nrow)}
  if(is.null(nrow)) { nrow = ceiling(n/ncol)}
  if(is.null(ncol)) { ncol = ceiling(n/nrow)}
  ## NOTE see n2mfrow in grDevices for possible alternative
  grid.newpage()
  pushViewport(viewport(layout=grid.layout(nrow,ncol) ) )
  ii.p <- 1
  for(ii.row in seq(1, nrow)){
    ii.table.row <- ii.row
    if(as.table) {ii.table.row <- nrow - ii.table.row + 1}
    for(ii.col in seq(1, ncol)){
      ii.table <- ii.p
      if(ii.p > n) break
      print(dots[[ii.table]], vp=vp.layout(ii.table.row, ii.col))
      ii.p <- ii.p + 1
    }
  }
}

robust <- function(model){ #This calculates the Huber-White Robust standard errors -- code from http://thetarzan.wordpress.com/2011/05/28/heteroskedasticity-robust-and-clustered-standard-errors-in-r/
    s <- summary(model)
    X <- model.matrix(model)
    u2 <- residuals(model)^2
    XDX <- 0

    for(i in 1:nrow(X)) {
        XDX <- XDX +u2[i]*X[i,]%*%t(X[i,])
    }

# inverse(X'X)
    XX1 <- solve(t(X)%*%X)

#Compute variance/covariance matrix
    varcovar <- XX1 %*% XDX %*% XX1

# Degrees of freedom adjustment
    dfc <- sqrt(nrow(X))/sqrt(nrow(X)-ncol(X))
```

```
67
68      stdh <- dfc*sqrt(diag(varcovar))
69
70      t <- model$coefficients/stdh
71      p <- 2*pnorm(-abs(t))
72      results <- cbind(model$coefficients, stdh, t, p)
73      dimnames(results) <- dimnames(s$coefficients)
74      results
75  }
76
77  ## Two functions for clustered standard errors below from: http://people.su.
        se/~ma/clustering.pdf -------
78
79  clx <-
80    function(fm, dfcw, cluster){
81      # R-codes (www.r-project.org) for computing
82      # clustered-standard errors. Mahmood Arai, Jan 26, 2008.
83
84      # The arguments of the function are:
85      # fitted model, cluster1 and cluster2
86      # You need to install libraries 'sandwich' and 'lmtest'
87
88      # reweighting the var-cov matrix for the within model
89      library(sandwich);library(lmtest)
90      M <- length(unique(cluster))
91      N <- length(cluster)
92      K <- fm$rank
93      dfc <- (M/(M-1))*((N-1)/(N-K))
94      uj  <- apply(estfun(fm),2, function(x) tapply(x, cluster, sum));
95      vcovCL <- dfc*sandwich(fm, meat=crossprod(uj)/N)*dfcw
96      coeftest(fm, vcovCL) }
97
98  mclx <-
99    function(fm, dfcw, cluster1, cluster2){
100     # R-codes (www.r-project.org) for computing multi-way
101     # clustered-standard errors. Mahmood Arai, Jan 26, 2008.
102     # See: Thompson (2006), Cameron, Gelbach and Miller (2006)
103     # and Petersen (2006).
104     # reweighting the var-cov matrix for the within model
105
106     # The arguments of the function are:
107     # fitted model, cluster1 and cluster2
108     # You need to install libraries 'sandwich' and 'lmtest'
109
110     library(sandwich);library(lmtest)
111     cluster12 = paste(cluster1,cluster2, sep="")
112     M1  <- length(unique(cluster1))
113     M2  <- length(unique(cluster2))
114     M12 <- length(unique(cluster12))
115     N   <- length(cluster1)
116     K   <- fm$rank
117     dfc1  <- (M1/(M1-1))*((N-1)/(N-K))
118     dfc2  <- (M2/(M2-1))*((N-1)/(N-K))
119     dfc12 <- (M12/(M12-1))*((N-1)/(N-K))
120     u1j   <- apply(estfun(fm), 2, function(x) tapply(x, cluster1,  sum))
121     u2j   <- apply(estfun(fm), 2, function(x) tapply(x, cluster2,  sum))
122     u12j  <- apply(estfun(fm), 2, function(x) tapply(x, cluster12, sum))
```

```
    vc1   <-  dfc1*sandwich(fm, meat=crossprod(u1j)/N )
    vc2   <-  dfc2*sandwich(fm, meat=crossprod(u2j)/N )
    vc12  <- dfc12*sandwich(fm, meat=crossprod(u12j)/N)
    vcovMCL <- (vc1 + vc2 - vc12)*dfcw
    coeftest(fm, vcovMCL)}

## Function to compute ols standard errors , robust, clustered...
## Based on http://diffuseprior.wordpress.com/2012/06/15/standard-robust-and-clustered-standard-errors-computed-in-r/
ols.hetero <- function(form, data, robust=FALSE, cluster=NULL,digits=3){
  r1 <- lm(form, data)
  if(length(cluster)!=0){
    data <- na.omit(data[,c(colnames(r1$model),cluster)])
    r1 <- lm(form, data)
  }
  X <- model.matrix(r1)
  n <- dim(X)[1]
  k <- dim(X)[2]
  if(robust==FALSE & length(cluster)==0){
    se <- sqrt(diag(solve(crossprod(X)) * as.numeric(crossprod(resid(r1))/(n-k))))
    res <- cbind(coef(r1),se)
  }
  if(robust==TRUE){
    u <- matrix(resid(r1))
    meat1 <- t(X) %*% diag(diag(crossprod(t(u)))) %*% X
    dfc <- n/(n-k)
    se <- sqrt(dfc*diag(solve(crossprod(X)) %*% meat1 %*% solve(crossprod(X))))
    res <- cbind(coef(r1),se)
  }
  if(length(cluster)!=0){
    clus <- cbind(X,data[,cluster],resid(r1))
    colnames(clus)[(dim(clus)[2]-1):dim(clus)[2]] <- c(cluster,"resid")
    m <- dim(table(clus[,cluster]))
    dfc <- (m/(m-1))*((n-1)/(n-k))
    uclust  <- apply(resid(r1)*X,2, function(x) tapply(x, clus[,cluster], sum))
    se <- sqrt(diag(solve(crossprod(X)) %*% (t(uclust) %*% uclust) %*% solve(crossprod(X)))*dfc)
    res <- cbind(coef(r1),se)
  }
  res <- cbind(res,res[,1]/res[,2],(1-pnorm(abs(res[,1]/res[,2])))*2)
  res1 <- matrix(as.numeric(sprintf(paste("%.",paste(digits,"f",sep=""),sep=""),res)),nrow=dim(res)[1])
  rownames(res1) <- rownames(res)
  colnames(res1) <- c("Estimate","Std. Error","t value","Pr(>|t|)")
  return(res1)
}
```

../util/are213–func.R