ARE213 Problem Set #2B

Peter Alstone & Frank Proulx

November 21, 2013

Part A: Preliminaries

(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 **??** 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.

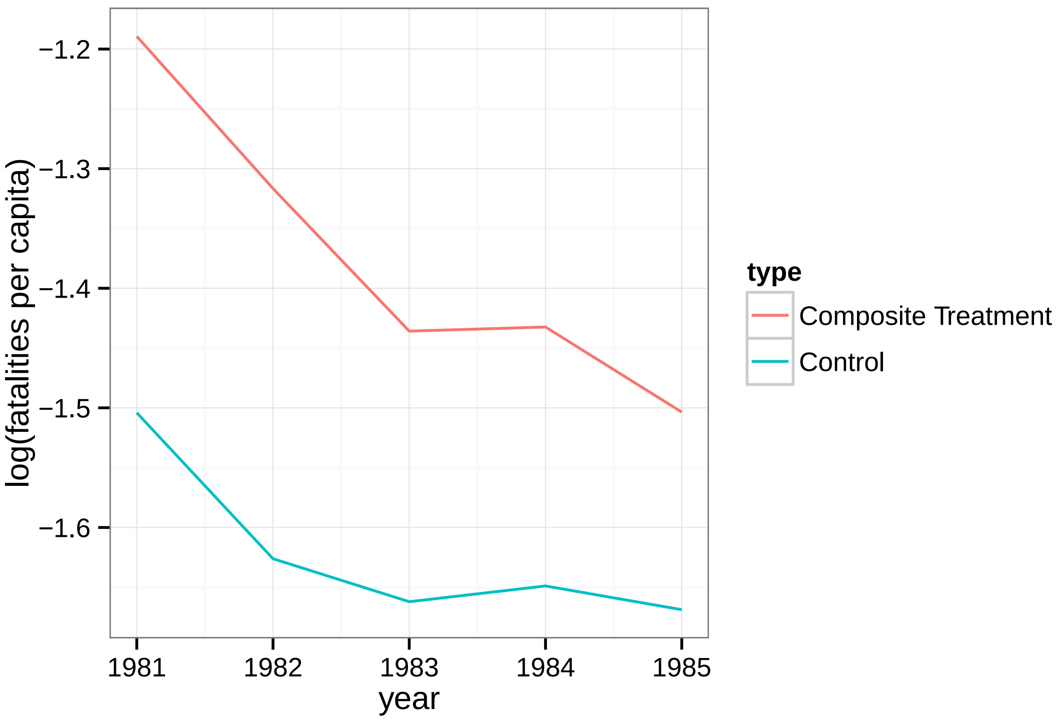


Figure 1: Trend in the dependent variable (log(fatalities per capita)) for the composite treatment state and the average of the control states.

(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 **??** below shows the distribution in the dependent variable

Fried green covariates and other stereotypes confirmed Tables **??** and **??** 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 **??**) 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.

Part B: Synthetic Controls

(i) 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.

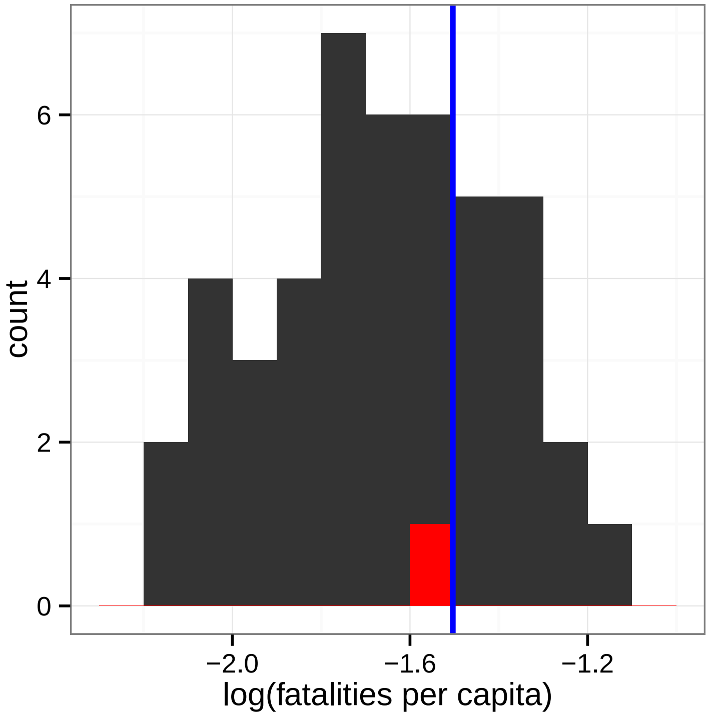


Figure 2: Distribution in traffic fatalities metric from 1985 for all control states with a vertical blue line indicating the value of the metric for the composite treatment state. The red block highlights the position of Alabama in the distribution. Alabama is the closest match to the composite treatment 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

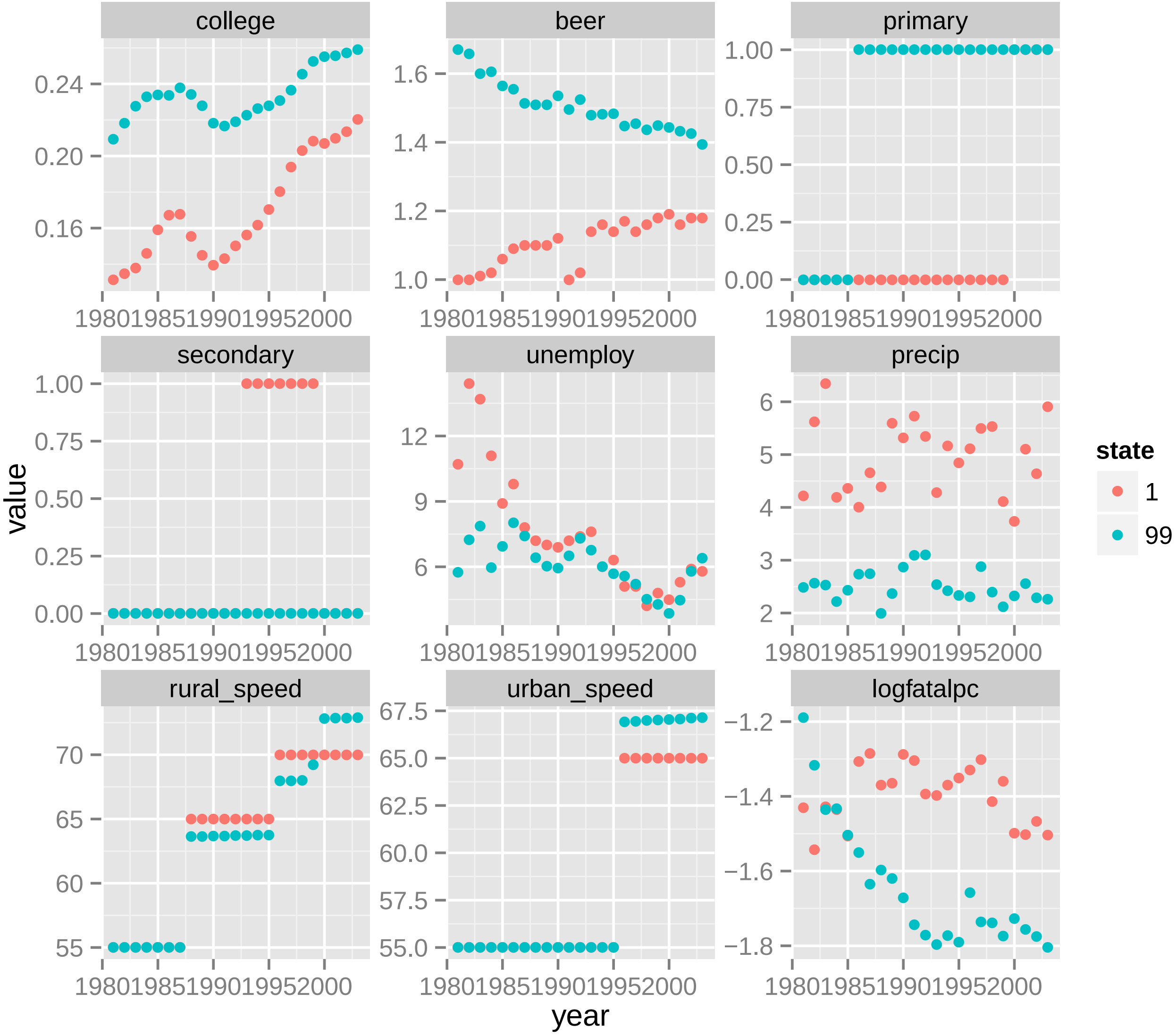


Figure 3: Trends in the covariate (and dependent) variables for the composite treatment state (99) and Alabama (1)

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 W that minimizes the distance between the treatment covariates (e.g. alcohol consumption, total VMT) and pre-intervention outcomes (log fatalities per capita) for the weighted control unit and the treatment unit. In particular, the following more formally defined criteria are sought (from Synth R package documentation):

- $\sum_{j=2}^{J+1} w_j^* \bar{Y}_j^{K_1} = \bar{Y}_1^{K_M}$, where j refers to the state, w_j is state j's weight, and $\bar{Y}_j^{K_m}$ denotes the pre-treatment outcome in state j in year m

- $\sum_{j=2}^{J+1} w_j^* U_j = U_1$, where U_i is a vector of covariates for state i

Pursuant to these criteria, the synthetic control method estimates the treatment effect as $\hat{\alpha}_{1t} = Y_{1t} - \sum_{j=2}^{J+1} w_j^* Y_{jt}$

The steps taken in this estimation by the Synth package are as follows:

1. Define a (k x 1) matrix (dubbed X_1) of the characteristics (covariates U_1 and pre-treatment outcomes $\bar{Y}_1^{K_m}$) of the treatment unit and a similar (k x J) matrix (dubbed X_J) for the control units.

2. Weight the control characteristics matrix with weight vector W.

3. Minimize the distance between the treatment unit characteristic matrix and the weighted control characteristics matrix with respect to the weighting matrix. Formally, that's $\min_W \sqrt{(X_1 - X_0W)'V(X_1 - X_0W)}$ where V is chosen by default to minimize the mean square error of the estimator.

Pros and Cons: The upsides to Synthetic control are that one can create a better match for the treatment unit than exists in reality and that it is a method that prevents issues of selection bias (i.e., it is possible to say, “I am using synthetic control” instead of needing to justify ex post the selection of particular units to match in classic differences in differences methods). Another nice feature of the method is the use of graphical placebo testing analysis for determination of the statistical power of results. It is an elegant and compelling way to approach error analysis. A potential methodological downside is that the method approaches a black box estimate that does not provide much intuition compared to other methods. This may manifest as a lack of trust in results from this method compared to those that are more straightforward to understand.

(ii) 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. Additionally, the presence of secondary seatbelt laws does not vary in the pre-treatment period so is also left out of the potential covariates.

Preferred specification: We identified that the following specifications were best for synthetic control analysis of this data:

- Covariates to include in pre-treatment "training" period: Full set of tractable and reasonable covariates. This includes alcohol consumption, VMT per capita, college educational attainment, precipitation, snowfall, and unemployment rate. We tried other combinations of covariates and found little influence on the result. We hoped a VMT-only specification would provide clarity but the gap in the pre-treatment period was biased compared to using the full set of covariates.

- Pre-treatment period: We use the full set of years available in the data, from 1981 to 1985, for the pre-treatment period. We considered dropping 1985 to avoid anticipation effects but this is not done for two reasons: first, it did not have noticeable impacts on the results (i.e., the divergence between treatment and synthetic control appears between 1985-1986 regardless of whether 1985 is included), and second, because there is a very short pre-treatment period available and we wished to maximize the support of the data.

Part C: ...but does it work?

(i) Gap between TU and synth control

We show a series of figures (**??** & **??**) for various specifications (including the preferred specification) below. Both the gaps (Synthetic unit - Treated unit) and the actual values have been plotted here.

The mean gap is about 0.15 on the log scale, which corresponds to approximately a 15% reduction in traffic fatalities per capita.

(ii) Gap between TU & preferred synth spec. and gap between each control state and its "placebo" treatment

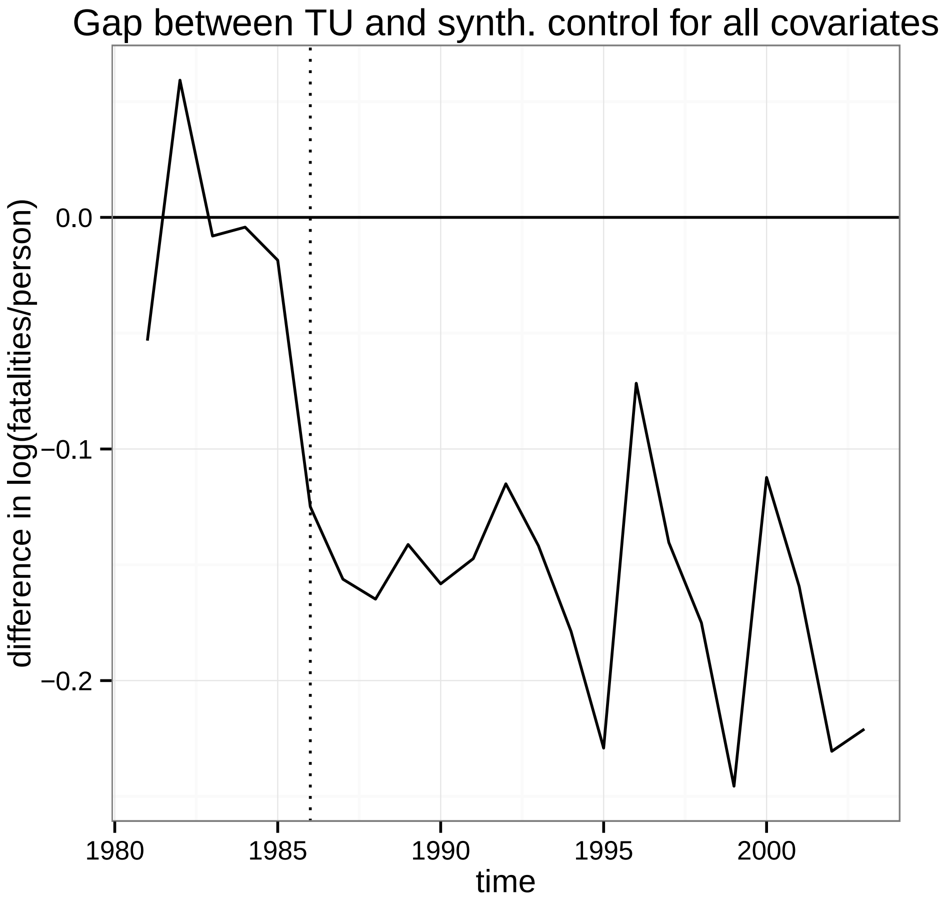
Graphical significance?

(iii) MSPE Ratios

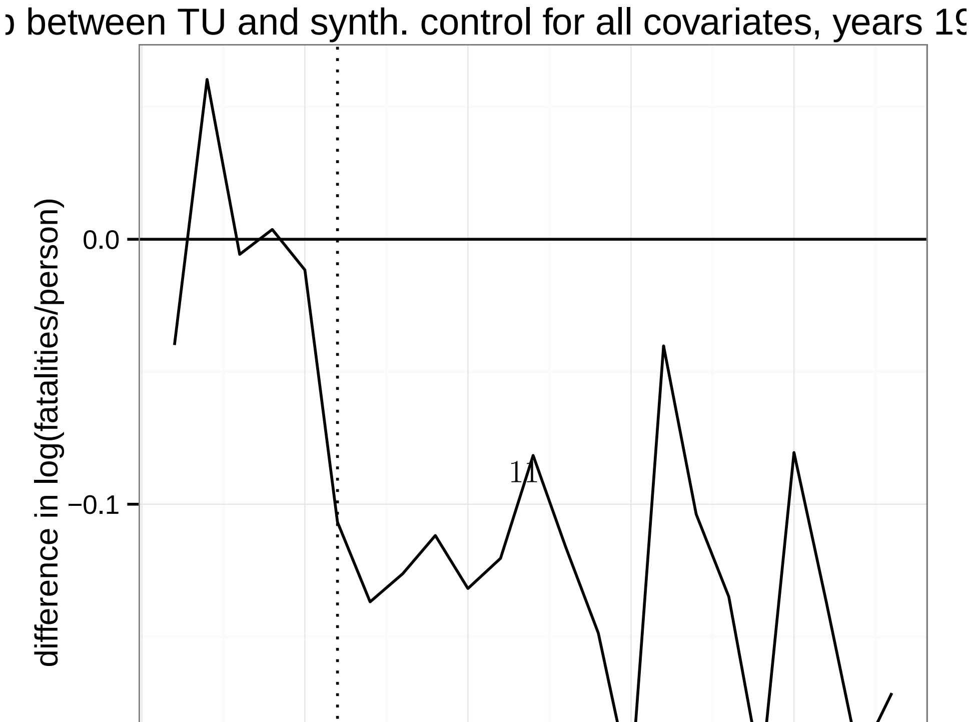
... Was it significant?? We find that the MSPE ratio for the TU is about 20, which is the second highest among the combined set of the TU and the control states. Florida (as a placebo) had a higher MSPE ratio, about 120. This sows seeds of doubt but we still find that the TU has a higher apparent effect then 93% of units.

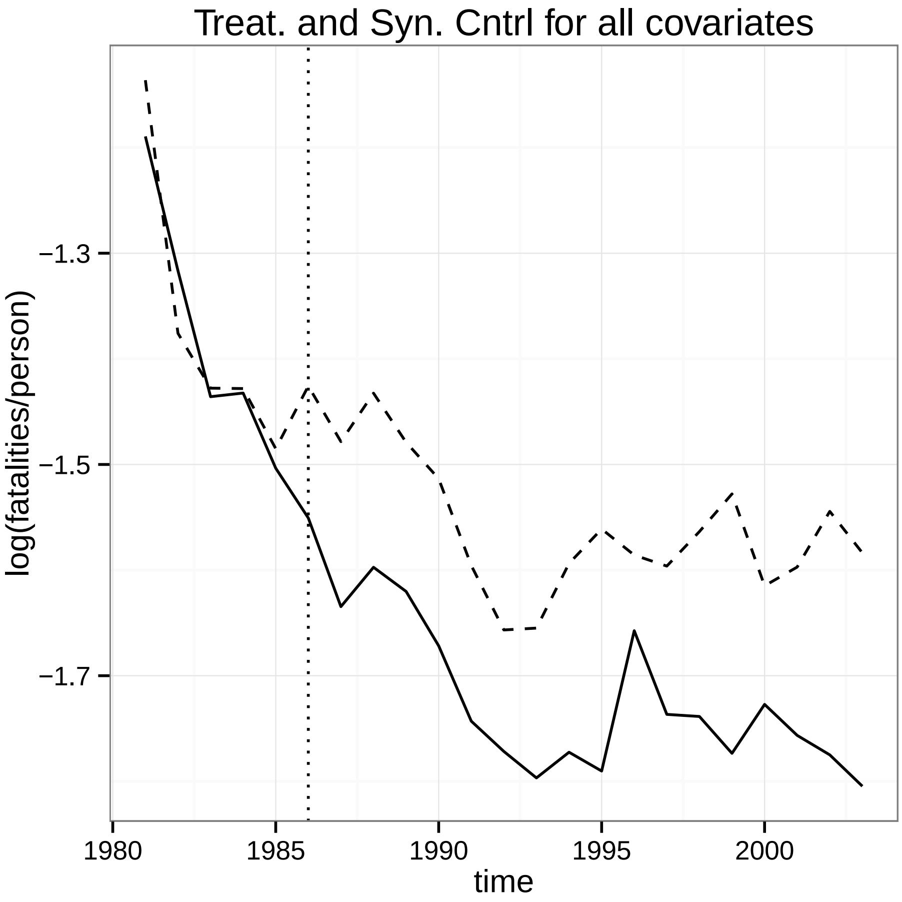
Part D: Compare with FE Model

Our central estimate for the impact of seatbelt laws using synthetic control methods is a 15% reduction in fatality rate. This is roughly double the estimates we found using a fixed effects model (8% with significance at a 0.05 level) in the previous assignment. It is notable that many of the alternative

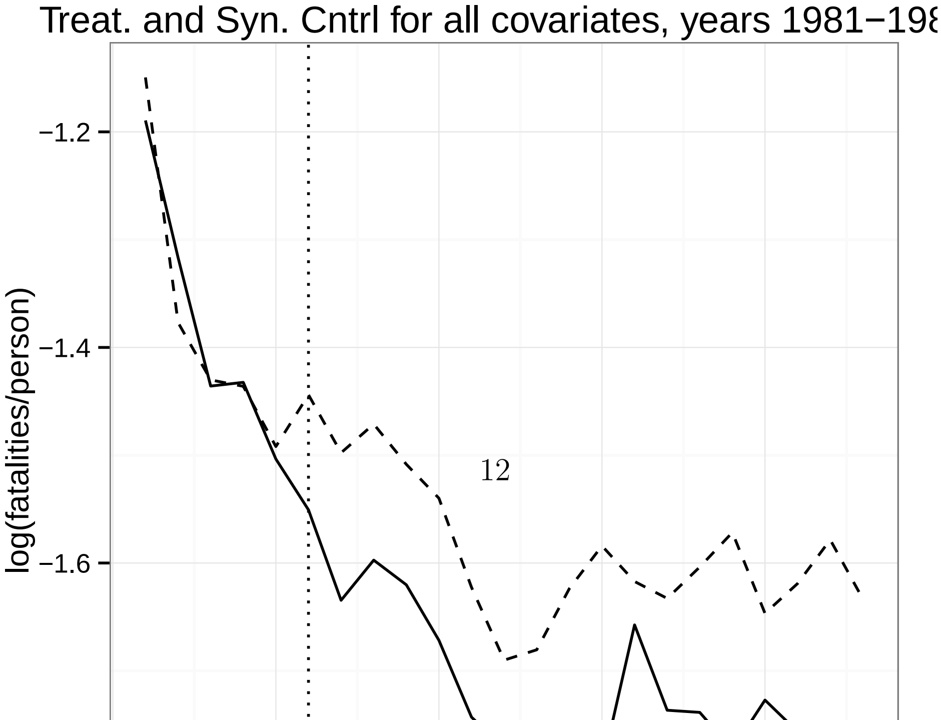


(a) All covariates, all times





(a) All covariates, all times



estimates for the coefficient on primary seatbelt laws had higher (but not significant) results closer to those we find with the synthetic controls method.

The differences may stem from a reduced sample of treatment states in the synthetic control method. Since only four states with primary seatbelt laws were included (out of nearly 20 that eventually have the laws) it could be the case that the analysis is biased.

Another source of difference is in the structural way the estimates are made. Where the fixed effects model compares pre and post treatment, the synthetic controls method compares only in the post-treatment period. The fixed effects model includes first and second order corrections for year which should address some of these issues, but to the extent that the trend in national-level drivers for traffic fatalities does not obey the quadratic functional form there are potential errors that are not present in the synthetic control method, since synthetic controls does not impose a linear function in the same way on the covariates.

Part E: Appendix: Code Listings

```
## Frank's wd
## setwd("/media/frank/Data/documents/school/berkeley/fall13/are213/are213/ps2")
## Peter's wd
# setwd("~/Google Drive/ERG/Classes/ARE213/are213/ps2")

library(foreign) #this is to read in Stata data
library(Hmisc)
library(psych)
library(stargazer)
library(ggplot2) # for neato plotting tools
library(plyr) # for nice data tools like ddply
library(car) # "companion for applied regression" - recode fxn, etc.
library(gmodels) #for Crosstabs
library(plm) # for panel data
library(Synth) #for synthetic control
library(reshape)

source("../util/are213-func.R")

ps2a.data <- read.dta('traffic_safety2.dta')
ps2a.datakey <- data.frame(var.name=names(ps2a.data), var.labels = attr(ps2a.data, "var.labels"))
ps2a.data$logfatalpc <- with(ps2a.data, log(fatalities/population))
ps2a.data$sqyears <- with(ps2a.data, year^2)

# make state labels dataframe
state.labels <- attr(ps2a.data, "label.table")
state.labels$state.name <- attr(state.labels$state_number,"names")
state.labels <- as.data.frame(state.labels)
state.labels$state <- state.labels$state_number
state.labels$state.name <- as.character(state.labels$state.name)
state.labels$state_number <- NULL
state.labels <- rbind(state.labels, c("TU", 99))



tu.label <- data.frame(state_number = 99, state.name = "TU")


## Problem 1 --------------------

## Part a -------------
## Subpart i ------------
                                        # Average pre-period log fatalities per cap.
pre.treatment.avg <- with(subset(ps2a.data, (state == 99 & primary == 0)), mean(logfatalpc))
pre.control.avg <- with(subset(ps2a.data, (state != 99 & primary == 0)), mean(logfatalpc))

                                        # Define state types
ps2a.data$type <- "Control"

for(i in 1:length(ps2a.data$year)){
  if(ps2a.data$state[i] == 99){
      ps2a.data$type[i] <- "Composite Treatment"}

  if(ps2a.data$state[i] %in% c(4,10,30,41)){
      ps2a.data$type[i] <- "Treatment State"}
}

use.rows <- which(ps2a.data$type != "Treatment State")

pre.period.data <- ddply(ps2a.data[use.rows,], .(year, type), summarize,
                         logfatalpc = mean(logfatalpc))

                                        # Plot of average fatality rates pre-1986.
preperiod.plot <- ggplot(data = subset(pre.period.data, year < 1986), aes(x = year, y = logfatalpc, color = type))
preperiod.plot <- preperiod.plot +
    geom_line() +
    theme_bw() +
    ylab("log(fatalities per capita)")

ggsave(filename = "img-p2b.logfatTrend.pdf", plot = preperiod.plot, width = 6, height = 4)

## Subpart ii
                                        # Compare logfatalpc in year before treatment

year.before <- ddply(subset(ps2a.data, year == 1985 & type != "Treatment State"), .(state), summarize,
                     type = type,
                     logfatalpc = logfatalpc)

target.fatal <- year.before$logfatalpc[which(year.before$type== "Composite Treatment")]
year.before$distance <- abs(year.before$logfatalpc - target.fatal)
year.before$distance[which(year.before$type== "Composite Treatment")] <- NA


                                        # plot all states and TU
pdf("img-p2b-compareStates.pdf", width = 4, height = 4)

ggplot(year.before, aes(logfatalpc)) +
    geom_histogram(binwidth=0.1) +
    geom_histogram(data=subset(year.before, state==1), aes(logfatalpc), binwidth=0.1, fill="red") +
    theme_bw() +
    xlab("log(fatalities per capita)") +
    geom_vline(aes(xintercept=target.fatal), size = 1, color = "blue")

dev.off()


best.yb.match <- which(year.before$distance == min(year.before$distance, na.rm=T))

print(paste("The state number for the closest year before match is", year.before$state[best.yb.match]))

                                        # The best match is Alabama.

                                        # Tables comparing Alabama to the composite treatment group
stargazer(subset(ps2a.data, state==99),
          out = "tab-ps2b-1a.tex",
          title = "Composite Treatment Group Summary",
          label = "tab:a21")

stargazer(subset(ps2a.data, state==1),
          out = "tab-ps2b-1b.tex",
          title = "Closest match for pre-policy fatalities: Alabama",
          label = "tab:a22")


# graphical comparison between states

prep.gg.compare <- melt.data.frame(subset(ps2a.data, state==99 | state ==1), id.vars = c("state","year"))

prep.gg.compare$value <- as.numeric(prep.gg.compare$value)
prep.gg.compare$state <- as.factor(prep.gg.compare$state)

vars.we.care.about <- c("beer", "college", "primary", "secondary", "unemploy", "logfatalpc", "rural_speed", "urban_speed", "precip")
rows.we.care.about <- which(prep.gg.compare$variable %in% vars.we.care.about)

```

```r
gg.compare.states <- ggplot(prep.gg.compare[rows.we.care.about,], aes(x=year
    , y=value, color=state)) +
    geom_point() +
    facet_wrap("variable", scales = "free")

pdf("img-ps2b-compareStatesFacets.pdf", width=7, height = 6)
gg.compare.states
dev.off()

## Part B - Synthetic control method


                                        # Identify which states can be "
                                            control" (i.e., those that never
                                             have a primary seatbelt law)

state.policy <- ddply(ps2a.data, .(state), summarize, primary.max = max(
    primary))
good.states <- which(state.policy$primary.max < 1)
control.states <- state.policy$state[good.states]
all.states <- c(control.states, 99)

## subpart ii


# add to dataset
ps2a.data$vmt_percapita <- with(ps2a.data, totalvmt / population)
ps2a.data <- join(ps2a.data, state.labels)

#helper function: extract useful data from synth results
get.plot.data <- function(synth.res, dataprep.res, label="unlabeled", type =
     "unknown"){
    out <- data.frame(time=dataprep.res$tag$time.plot)

    synthetic.trend <- dataprep.res$Y0plot %*% synth.res$solution.w
    treatment.trend <- dataprep.res$Y1plot
    gap <- treatment.trend - synthetic.trend

    out$synth <- as.numeric(synthetic.trend)
    out$treat <- as.numeric(treatment.trend)
    out$gap <- as.numeric(gap)
    out$spe <- out$gap^2
    out$label <- label
    out$type <- type


    return(out)
}

#helper function: run dataprep, synth, and get.plot.data for a set of inputs
    .
run.syn <- function(predictor.set, time.prior, treatment.state = std.
    treatment.state, control.states, label = "unlabeled", type = "unknown"){

# synthetic controls dataprep with specified set of tractable predictors
    dataprep.results <- dataprep(foo = ps2a.data,
                                 predictors = predictor.set,
                                 predictors.op = c("mean"),
                                 dependent = "logfatalpc",
                                 unit.variable = "state",
                                 time.variable = "year",
                                 treatment.identifier = treatment.state,
                                 controls.identifier = control.states,
                                 time.predictors.prior = time.prior,
                                 time.optimize.ssr = time.prior,
                                 time.plot = c(1981:2003),
                                 unit.names.variable = "state.name"
                                 )

    synth.results <- synth(data.prep.obj = dataprep.results)

    output <- get.plot.data(synth.results, dataprep.results, label, type)

    return(output)
}

## helper function: makes a gap plot so we can get them to look consistent
make.gap.plot <- function(syntheticresults, finame, desc){
    gap.plot <- ggplot(data = syntheticresults, aes(y = gap, x = time))
    gap.plot <- gap.plot +
        geom_path() +
            geom_vline(xintercept = 1986, linetype = 'dotted') +
                theme_bw() + geom_hline(yintercept = 0) + ylab("difference in log(fatalities/person)") +
                    labs(title = paste("Gap between TU and synth. control for", desc))
    ggsave(filename = paste0('img-gap-', finame, '.pdf'), plot = gap.plot, width=5, height = 5)

    split.plot <- ggplot(data = syntheticresults, aes(y = treat, x = time))
    split.plot <- split.plot +
        geom_path() +
            geom_path(aes(y = synth), linetype = 'dashed') +
                geom_vline(xintercept = 1986, linetype = 'dotted') +
                        theme_bw() + ylab('log(fatalities/person)') +
                            labs(title = paste("Treat. and Syn. Cntrl for", desc))
    ggsave(filename = paste0('img-split-', finame, '.pdf'), plot = split.plot, width=5, height = 5)

    return(gap.plot)
}


                                        # default entries to run.syn function
std.treatment.state <- 99
full.predictor.set <- c("college", "precip", "snow32", "beer", "vmt_percapita", "unemploy")
full.time.prior <- c(1981:1985)

## Part C: Pretty pictures
## Part (i)
```

```
full.synthesis <- run.syn(full.predictor.set, full.time.prior, 99, control.states)
full.gap.plot <- make.gap.plot(full.synthesis, ’full’, ’all covariates’)

full.synthesis1984 <- run.syn(full.predictor.set, c(1981:1984), 99, control.states)
full.1984.gap.plot <- make.gap.plot(full.synthesis1984, ’full1984’, ’all covariates, years 1981-1984’)

vmt.synthesis <- run.syn("vmt_percapita", full.time.prior, 99, control.states)
vmt.gap.plot <- make.gap.plot(vmt.synthesis, ’vmt’, ’VMT per capita covariate’)

# Script to generate a placebo test plot
## Part (ii)

placebo.test <- run.syn(full.predictor.set, full.time.prior, treatment.state = 99, control.states, label = 99, type = "treatment")

for(state in control.states){
    updated.control <- control.states[which(control.states!=state)]
    placebo.additional.result <- run.syn(full.predictor.set, full.time.prior, state, updated.control, label = state, type = "placebo")
    placebo.test <- rbind(placebo.test, placebo.additional.result)
    print(paste("Finished with state number", state, "and you should be patient for the rest to finish :)"))
}

placebo.test$treated <- (placebo.test$time > 1985)

MPSEs <- ddply(placebo.test, .(label), summarize,
               preMPSE = mean(spe[!treated]),
               postMPSE = mean(spe[treated]))
MPSEs$ratio <- with(MPSEs, postMPSE/preMPSE)


# Exclusion threshold for placebo plot based on
preThreshold <- 50 #exclude states with pre-intervention MSPE greater than x times that for TU
tu.mspe <- MPSEs$preMPSE[which(MPSEs$label==99)]
include.states <- MPSEs$label[which(MPSEs$preMPSE < tu.mspe * preThreshold)]
include.states.rows <- which(placebo.test$label %in% include.states)

ex.number <- length(control.states) - length(include.states) -1
in.number <- length(include.states)

# plot all the lines
placeboTest <- ggplot(placebo.test[include.states.rows,], aes(time, gap, group=label))
placeboTest <- placeboTest +
    geom_line(aes(color=type))+
    geom_vline(aes(xintercept = 1985), linetype="dotted") +
    theme_bw() +
    xlab("year") +
    ylab("Estimated gap between treatment and synthetic control") +
    scale_color_brewer(palette = "Paired") +
    geom_hline(yintercept = 0) +
    ggtitle(paste0("Placebo test plot with ",ex.number," control states excluded that have \n >", preThreshold, "x divergence from TU pre-treatment MSPE\n(includes ", in.number, " control states)"))
ggsave(paste0("img-placeboTest", preThreshold, ".pdf"), plot = placeboTest, width = 6, height = 5)


## Part(iii)


MPSE.plot <- ggplot(MPSEs, aes(ratio))
MPSE.plot <-  MPSE.plot +
  geom_histogram() +
  geom_histogram(data=subset(MPSEs, label==99), aes(ratio), fill="blue") +
  theme_bw() +
  annotate(geom="text", x=50, y=3, label="Blue indicates TU") +
  xlab("MSPE ratio (post:pre)")

ggsave(filename = "img-mspeRatio.pdf", plot = MPSE.plot, width = 5, height = 5)

# # Visual exploration of plots....only works if you have the native dataprep and synth objects.  We would rather make our own plots :)
# gaps.plot(seatbelts.synth, syn.data.full)
# path.plot(seatbelts.synth, syn.data.full, Ylim = c(-1.3, -2))

# # Tables for synthetic controls results
# syn.table <- synth.tab(seatbelts.synth, syn.data.full,3)
```

./ps2b.r

```
# Econometrics helper functions for [R]
#
# Peter Alstone and Frank Proulx
# 2013
# version 1
# contact: peter.alstone AT gmail.com

# Category: Data Management -------------


# Category: Data Analysis ----------------

# Function: Find adjusted R^2 for subset of data
# 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))

    stdh <- dfc*sqrt(diag(varcovar))

    t <- model$coefficients/stdh
    p <- 2*pnorm(-abs(t))
    results <- cbind(model$coefficients, stdh, t, p)
    dimnames(results) <- dimnames(s$coefficients)
    results
}

```

```
## Two functions for clustered standard errors below from: http://people.su.se/˜ma/clustering.pdf -------

clx <-
  function(fm, dfcw, cluster){
    # R-codes (www.r-project.org) for computing
    # clustered-standard errors. Mahmood Arai, Jan 26, 2008.

    # The arguments of the function are:
    # fitted model, cluster1 and cluster2
    # You need to install libraries ‘sandwich’ and ‘lmtest’

    # reweighting the var-cov matrix for the within model
    library(sandwich);library(lmtest)
    M <- length(unique(cluster))
    N <- length(cluster)
    K <- fm$rank
    dfc <- (M/(M-1))*((N-1)/(N-K))
    uj  <- apply(estfun(fm),2, function(x) tapply(x, cluster, sum));
    vcovCL <- dfc*sandwich(fm, meat=crossprod(uj)/N)*dfcw
    coeftest(fm, vcovCL) }

mclx <-
  function(fm, dfcw, cluster1, cluster2){
    # R-codes (www.r-project.org) for computing multi-way
    # clustered-standard errors. Mahmood Arai, Jan 26, 2008.
    # See: Thompson (2006), Cameron, Gelbach and Miller (2006)
    # and Petersen (2006).
    # reweighting the var-cov matrix for the within model

    # The arguments of the function are:
    # fitted model, cluster1 and cluster2
    # You need to install libraries ‘sandwich’ and ‘lmtest’

    library(sandwich);library(lmtest)
    cluster12 = paste(cluster1,cluster2, sep="")
    M1  <- length(unique(cluster1))
    M2  <- length(unique(cluster2))
    M12 <- length(unique(cluster12))
    N   <- length(cluster1)
    K   <- fm$rank
    dfc1  <- (M1/(M1-1))*((N-1)/(N-K))
    dfc2  <- (M2/(M2-1))*((N-1)/(N-K))
    dfc12 <- (M12/(M12-1))*((N-1)/(N-K))
    u1j   <- apply(estfun(fm), 2, function(x) tapply(x, cluster1,  sum))
    u2j   <- apply(estfun(fm), 2, function(x) tapply(x, cluster2,  sum))
    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