ARE 213 Problem Set 2B

Becky Cardinali, Yuen Ho, Sara Johns, and Jacob Lefler

Due 11/16/2020

Question 1

We first estimate an event study specification.

(a) First determine the minimum and maximum event time values that you can estimate in this data set. Code up a separate event time indicator for each possible value of event time in the data set. Estimate an event study regression using all the event time indicators. What happens?

The data set contains data for each year in $[1981, 2003]$. Therefore, the minimum number of event time values we can estimate in this data set is 23 $(2003 - 1981 + 1)$, which would be the case where all states are treated in the same year, all states are treated during all years in the data set, or all states are untreated during all years in the data set.

The maximum event time value we can estimate in this data set is 47?? $(2003 - 1981 + 1) * 2$, which would be the case where at least one state was treated for the entire sample (event time 1 to 23) and at least one state was never treated during the sample (event time -23 to -1) (23 + 23 + 1 for event time 0)

We have a single treated state ($s = 1$ when the state has a primary seat belt law) and a single control state ($s = 0$ when a state does not have a primary seat belt law). So for an event study regression using all the event time indicators, we will estimate the regression:

$$Y_{ist} = \alpha + \sum_{j=-T_0}^{T-T_0} \tau_j D_{jst} + \gamma_s + \delta_t + \epsilon_{st} + u_{ist}$$

where T_0 is the period just prior to treatment, and D_{jst} is an indicator function for period t falling j periods after T_0 in the treated state (i.e., $\mathbf{1}(t - T_0 = j) * \mathbf{1}(s = 1)$), so index j is "event time."

```
# Becky's notes to be deleted later
# what if the primary seat belt law was taken away? i don't think it would show up in event time indicator, b
# how to code event time for places that never get treated? doesn't matter because they won't show up in the
# how to code event time for places that were treated throughout the sample? it's luckily not the case in our
# work out case for completely treated? think it won't work above because of second for loop with i in 0:1
# doing logs like in pset 1, what about controls, 1+ for constant?
# since the sum of the event time indicators is collinear with the treated state's fixed effect. ???
# do you need dummy_cols if you use fixef?


# separate event time indicator for each possible value of event time in the data set

# create event time variable
traffic <- traffic[, event_time := NA]

# iterate through all states (this includes state 99)
for(s in unique(traffic$state)){

  # data frame for just state s
  temp <- subset(traffic, state == s)
  # make sure it's in ascending order by year
```

```r
  temp <- temp[order(year), ]

  # find the row # and year corresponding to the first occurrence of primary == 1 in the dataframe for just t
  row_of_first_occ <- match(1, temp$primary)
  first_treated_year <- traffic[match(1, temp$primary), year]

  # case where a state was never treated in the sample (row_of_first_occ == NA)
  if(is.na(row_of_first_occ)){
    k <- -1
    for(i in 23:1){
       # year 2003 corresponds to event time = -1, year 2002 corresponds to event time = -2, etc.
      setDT(traffic)[, event_time := ifelse(state == s & year == (2004 + k), k, event_time)]
      k <- k - 1
    }
  }

  # case where a state was treated in the sample (row_of_first_occ > 1)
  # no cases of a state being treated during the entire sample
  else{
    # for the first year with primary == 1, make event time = 0
    setDT(traffic)[, event_time := ifelse(state == s & year == first_treated_year, 0, event_time)]

    k <- 1
    for(i in (row_of_first_occ+1):23){
      # year after event time 0 corresponds to event time 1, 2 years after event time 0 corresponds to event
      setDT(traffic)[, event_time := ifelse(state == s & year == (first_treated_year + k), k, event_time)]
      k <- k + 1
      }

    k <- -1
    for(i in (row_of_first_occ-1):1){
      # year before event time 0 corresponds to event time -1, 2 years before event time 0 corresponds to eve
      setDT(traffic)[, event_time := ifelse(state == s & year == (first_treated_year + k), k, event_time)]
      k <- k - 1
      }
  }
}

# remove 99 for this part of the analysis
traf_pt1 <- traffic[!(state==99),]

# make separate event time indicator for each possible value of event time in the data set
traf_pt1 <- traf_pt1[,dummy_cols(traf_pt1, select_columns = c("event_time"))]

# get ever treated dummy
traf_pt1[,treat:=max(primary), by = state]
# interact with ever treated
event_time_cols <- names(traf_pt1[,`event_time_-1`:`event_time_19`]) # get list of event time columns
# multiply ever treated by all event time columns
traf_pt1 <- traf_pt1[, (event_time_cols) := lapply(.SD, function(x) x * traf_pt1$treat), .SDcols = event_time

# create y variable
traf_pt1[, ln_fat_pc := log((fatalities/population))]

# make table with just the variables for the regression (makes writing out regression equation easier)
traf_reg <- traf_pt1[,c("ln_fat_pc", "state", "year",..event_time_cols)]
traf_reg[,state:=factor(state)]
traf_reg[,year:=factor(year)]
```

```r
# event study regression using all the event time indicators
es1 <- lm(ln_fat_pc ~ .-`event_time_-23`, data = traf_reg)
es1_cluster <- vcovCL(es1, type = "HC1", cluster=traf_reg$county) # get cluster cov-var
coeftest(es1, vcov. = es1_cluster)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)      -1.481476   0.052405 -28.2698 < 2.2e-16 ***
## state2            0.293631   0.046306   6.3411 3.457e-10 ***
## state3            0.241346   0.045047   5.3576 1.048e-07 ***
## state4           -0.471998   0.034549 -13.6616 < 2.2e-16 ***
## state5           -0.072784   0.046330  -1.5710 0.1165021
## state6           -0.692988   0.034146 -20.2950 < 2.2e-16 ***
## state7           -0.337649   0.037105  -9.0999 < 2.2e-16 ***
## state8            0.147233   0.044635   3.2986 0.0010060 **
## state9           -0.120736   0.028195  -4.2822 2.030e-05 ***
## state10          -0.301411   0.040653  -7.4143 2.617e-13 ***
## state11           0.250356   0.043635   5.7376 1.275e-08 ***
## state12          -0.339519   0.043814  -7.7490 2.282e-14 ***
## state13          -0.376337   0.028864 -13.0382 < 2.2e-16 ***
## state14           0.019534   0.045503   0.4293 0.6677986
## state15           0.171003   0.047139   3.6277 0.0003006 ***
## state16          -0.140850   0.034241  -4.1135 4.220e-05 ***
## state17          -0.707101   0.055900 -12.6494 < 2.2e-16 ***
## state18          -0.572411   0.028596 -20.0175 < 2.2e-16 ***
## state19          -0.100564   0.045869  -2.1924 0.0285791 *
## state20          -0.481306   0.029257 -16.4512 < 2.2e-16 ***
## state21          -0.315799   0.045065  -7.0076 4.470e-12 ***
## state22           0.108783   0.048494   2.2432 0.0251014 *
## state23           0.482778   0.052089   9.2684 < 2.2e-16 ***
## state24           0.395685   0.049589   7.9793 4.039e-15 ***
## state25          -0.084264   0.033854  -2.4890 0.0129723 *
## state26          -0.165773   0.053372  -3.1060 0.0019502 **
## state27          -0.057434   0.046904  -1.2245 0.2210522
## state28          -0.319373   0.056729  -5.6298 2.347e-08 ***
## state29          -0.830367   0.032652 -25.4310 < 2.2e-16 ***
## state30           0.299208   0.032006   9.3486 < 2.2e-16 ***
## state31           0.197036   0.054212   3.6346 0.0002927 ***
## state32          -0.768237   0.033455 -22.9633 < 2.2e-16 ***
## state33          -0.272984   0.044547  -6.1281 1.281e-09 ***
## state34          -0.112755   0.038486  -2.9298 0.0034697 **
## state35          -0.276773   0.032194  -8.5971 < 2.2e-16 ***
## state36          -0.277829   0.043313  -6.4144 2.184e-10 ***
## state37          -0.727848   0.052829 -13.7774 < 2.2e-16 ***
## state38           0.359231   0.045929   7.8215 1.330e-14 ***
## state39           0.176629   0.053437   3.3054 0.0009825 ***
## state40           0.236849   0.045094   5.2523 1.837e-07 ***
## state41          -0.136543   0.034029  -4.0126 6.457e-05 ***
## state42          -0.084894   0.047192  -1.7989 0.0723340 .
## state43          -0.192981   0.044786  -4.3089 1.804e-05 ***
## state44          -0.045195   0.056491  -0.8000 0.4238833
## state45          -0.601920   0.035526 -16.9432 < 2.2e-16 ***
## state46          -0.175266   0.043983  -3.9848 7.247e-05 ***
## state47           0.214710   0.047020   4.5663 5.586e-06 ***
## state48           0.562852   0.054361  10.3539 < 2.2e-16 ***
## year1982         -0.123497   0.031070  -3.9748 7.555e-05 ***
```

```
## year1983         -0.162298   0.028254  -5.7443 1.227e-08 ***
## year1984         -0.147374   0.029643  -4.9717 7.816e-07 ***
## year1985         -0.166030   0.030150  -5.5068 4.654e-08 ***
## year1986         -0.113736   0.028009  -4.0606 5.279e-05 ***
## year1987         -0.115232   0.029020  -3.9708 7.680e-05 ***
## year1988         -0.101326   0.028936  -3.5017 0.0004828 ***
## year1989         -0.160381   0.029349  -5.4646 5.865e-08 ***
## year1990         -0.180792   0.026994  -6.6974 3.546e-11 ***
## year1991         -0.251988   0.026358  -9.5603 < 2.2e-16 ***
## year1992         -0.308342   0.026995 -11.4223 < 2.2e-16 ***
## year1993         -0.308969   0.026201 -11.7922 < 2.2e-16 ***
## year1994         -0.310225   0.027769 -11.1715 < 2.2e-16 ***
## year1995         -0.284994   0.027926 -10.2052 < 2.2e-16 ***
## year1996         -0.297307   0.027915 -10.6505 < 2.2e-16 ***
## year1997         -0.281148   0.028212  -9.9654 < 2.2e-16 ***
## year1998         -0.298604   0.028907 -10.3297 < 2.2e-16 ***
## year1999         -0.301177   0.030505  -9.8730 < 2.2e-16 ***
## year2000         -0.323983   0.029777 -10.8803 < 2.2e-16 ***
## year2001         -0.315712   0.029494 -10.7042 < 2.2e-16 ***
## year2002         -0.297747   0.030886  -9.6402 < 2.2e-16 ***
## year2003         -0.307209   0.032814  -9.3621 < 2.2e-16 ***
## `event_time_-1`   0.292948   0.036587   8.0068 3.276e-15 ***
## `event_time_-2`   0.298072   0.039332   7.5783 8.008e-14 ***
## `event_time_-3`   0.322481   0.036214   8.9050 < 2.2e-16 ***
## `event_time_-4`   0.326763   0.046019   7.1007 2.363e-12 ***
## `event_time_-5`   0.315484   0.039248   8.0381 2.578e-15 ***
## `event_time_-6`   0.338474   0.040859   8.2840 3.826e-16 ***
## `event_time_-7`   0.331255   0.040670   8.1449 1.133e-15 ***
## `event_time_-8`   0.309575   0.042023   7.3668 3.670e-13 ***
## `event_time_-9`   0.294626   0.046710   6.3076 4.260e-10 ***
## `event_time_-10`  0.310506   0.044164   7.0307 3.817e-12 ***
## `event_time_-11`  0.323465   0.052849   6.1206 1.340e-09 ***
## `event_time_-12`  0.316051   0.050748   6.2279 6.970e-10 ***
## `event_time_-13`  0.341155   0.044232   7.7128 2.984e-14 ***
## `event_time_-14`  0.311298   0.044354   7.0184 4.152e-12 ***
## `event_time_-15`  0.352605   0.050830   6.9370 7.215e-12 ***
## `event_time_-16`  0.354292   0.061290   5.7806 9.959e-09 ***
## `event_time_-17`  0.346026   0.054186   6.3859 2.613e-10 ***
## `event_time_-18`  0.285088   0.086418   3.2989 0.0010050 **
## `event_time_-19`  0.309098   0.086906   3.5567 0.0003933 ***
## `event_time_-20`  0.369655   0.083251   4.4402 9.993e-06 ***
## `event_time_-21`  0.417160   0.068054   6.1298 1.268e-09 ***
## `event_time_-22`  0.138458   0.054935   2.5204 0.0118786 *
## event_time_0      0.262341   0.036045   7.2781 6.874e-13 ***
## event_time_1      0.248420   0.037058   6.7035 3.408e-11 ***
## event_time_2      0.246392   0.034258   7.1923 1.253e-12 ***
## event_time_3      0.246607   0.031438   7.8441 1.122e-14 ***
## event_time_4      0.227821   0.038205   5.9631 3.436e-09 ***
## event_time_5      0.212652   0.038159   5.5728 3.229e-08 ***
## event_time_6      0.196476   0.043208   4.5472 6.107e-06 ***
## event_time_7      0.217299   0.034969   6.2140 7.589e-10 ***
## event_time_8      0.161217   0.046741   3.4491 0.0005860 ***
## event_time_9      0.178955   0.044438   4.0271 6.078e-05 ***
## event_time_10     0.177224   0.033369   5.3110 1.345e-07 ***
## event_time_11     0.159365   0.045213   3.5248 0.0004432 ***
## event_time_12     0.144271   0.034009   4.2421 2.421e-05 ***
## event_time_13     0.152186   0.043083   3.5324 0.0004307 ***
## event_time_14     0.157418   0.046238   3.4045 0.0006892 ***
## event_time_15     0.150423   0.036436   4.1285 3.959e-05 ***
```

```
## event_time_16      0.097341   0.036615   2.6585 0.0079754 **
## event_time_17      0.084208   0.034808   2.4192 0.0157340 *
## event_time_18      0.014412   0.026227   0.5495 0.5827713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# lm chooses to remove event time 19
```

When we estimate an event study regression using all the event time indicators, the following message pops up: "The variable 'event_time' has been removed because of collinearity." ? This makes sense because the sum in $Y_{ist} = \alpha + \sum_{j=-T_0}^{T-T_0} \tau_j D_{jst} + \gamma_s + \delta_t + \epsilon_{st} + u_{ist}$ contains T dummy variables, which fully saturates event time and leads to the dummy variable trap since the sum of the event time indicators is collinear with the treated state's fixed effect.

(b) Estimate another event study regression using all the event time indicators save one that you choose to omit. Generate a plot of the event study coefficients.

```
# omit event_time -1
es2 <- lm(ln_fat_pc ~ .-`event_time_-1`-`event_time_-23`, data = traf_reg)
es2_cluster <- vcovCL(es2, type = "HC1", cluster=traf_reg$county) # get cluster cov-var
coeftest(es2, vcov. = es2_cluster)
```

```
##
## t test of coefficients:
##
##                    Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)      -1.18852823  0.04315851 -27.5387 < 2.2e-16 ***
## state2            0.00068287  0.03944322   0.0173 0.9861906
## state3           -0.05160204  0.03758614  -1.3729 0.1700931
## state4           -0.47199815  0.03454927 -13.6616 < 2.2e-16 ***
## state5           -0.36573129  0.03888297  -9.4059 < 2.2e-16 ***
## state6           -0.69298817  0.03414577 -20.2950 < 2.2e-16 ***
## state7           -0.33764872  0.03710467  -9.0999 < 2.2e-16 ***
## state8           -0.14571434  0.03705315  -3.9326 8.988e-05 ***
## state9           -0.12073629  0.02819516  -4.2822 2.030e-05 ***
## state10          -0.30141109  0.04065288  -7.4143 2.617e-13 ***
## state11          -0.04259129  0.03551435  -1.1993 0.2307092
## state12          -0.63246693  0.03571274 -17.7098 < 2.2e-16 ***
## state13          -0.37633715  0.02886428 -13.0382 < 2.2e-16 ***
## state14          -0.27341338  0.03796868  -7.2010 1.179e-12 ***
## state15          -0.12194495  0.03977059  -3.0662 0.0022266 **
## state16          -0.14084972  0.03424113  -4.1135 4.220e-05 ***
## state17          -1.00004837  0.05038377 -19.8486 < 2.2e-16 ***
## state18          -0.57241139  0.02859551 -20.0175 < 2.2e-16 ***
## state19          -0.39351183  0.03849212 -10.2232 < 2.2e-16 ***
## state20          -0.48130606  0.02925651 -16.4512 < 2.2e-16 ***
## state21          -0.60874681  0.03727425 -16.3316 < 2.2e-16 ***
## state22          -0.18416420  0.04124749  -4.4649 8.930e-06 ***
## state23           0.18983028  0.04585490   4.1398 3.771e-05 ***
## state24           0.10273703  0.04264025   2.4094 0.0161603 *
## state25          -0.08426371  0.03385411  -2.4890 0.0129723 *
## state26          -0.45872087  0.04753658  -9.6498 < 2.2e-16 ***
## state27          -0.35038152  0.03987355  -8.7873 < 2.2e-16 ***
## state28          -0.61232052  0.05127983 -11.9408 < 2.2e-16 ***
## state29          -0.83036675  0.03265178 -25.4310 < 2.2e-16 ***
## state30           0.29920827  0.03200555   9.3486 < 2.2e-16 ***
## state31          -0.09591159  0.04705211  -2.0384 0.0417732 *
## state32          -0.76823709  0.03345494 -22.9633 < 2.2e-16 ***
## state33          -0.56593216  0.03603110 -15.7068 < 2.2e-16 ***
```

```
## state34          -0.11275492  0.03848584  -2.9298 0.0034697 **
## state35          -0.27677296  0.03219383  -8.5971 < 2.2e-16 ***
## state36          -0.57077677  0.03516188 -16.2328 < 2.2e-16 ***
## state37          -1.02079569  0.04630771 -22.0438 < 2.2e-16 ***
## state38           0.06628298  0.03877033   1.7096 0.0876465 .
## state39          -0.11631872  0.04520789  -2.5730 0.0102271 *
## state40          -0.05609868  0.03756671  -1.4933 0.1356742
## state41          -0.13654335  0.03402881  -4.0126 6.457e-05 ***
## state42          -0.37784199  0.03881823  -9.7336 < 2.2e-16 ***
## state43          -0.48592913  0.03689519 -13.1705 < 2.2e-16 ***
## state44          -0.33814227  0.04690407  -7.2092 1.113e-12 ***
## state45          -0.60191952  0.03552572 -16.9432 < 2.2e-16 ***
## state46          -0.46821373  0.03561410 -13.1469 < 2.2e-16 ***
## state47          -0.07823720  0.03999711  -1.9561 0.0507368 .
## state48           0.26990388  0.04845499   5.5702 3.276e-08 ***
## year1982         -0.12349654  0.03107026  -3.9748 7.555e-05 ***
## year1983         -0.16229825  0.02825376  -5.7443 1.227e-08 ***
## year1984         -0.14737399  0.02964267  -4.9717 7.816e-07 ***
## year1985         -0.16602964  0.03015003  -5.5068 4.654e-08 ***
## year1986         -0.11373574  0.02800928  -4.0606 5.279e-05 ***
## year1987         -0.11523167  0.02901985  -3.9708 7.680e-05 ***
## year1988         -0.10132644  0.02893598  -3.5017 0.0004828 ***
## year1989         -0.16038054  0.02934874  -5.4646 5.865e-08 ***
## year1990         -0.18079205  0.02699438  -6.6974 3.546e-11 ***
## year1991         -0.25198766  0.02635768  -9.5603 < 2.2e-16 ***
## year1992         -0.30834214  0.02699473 -11.4223 < 2.2e-16 ***
## year1993         -0.30896933  0.02620123 -11.7922 < 2.2e-16 ***
## year1994         -0.31022529  0.02776924 -11.1715 < 2.2e-16 ***
## year1995         -0.28499365  0.02792619 -10.2052 < 2.2e-16 ***
## year1996         -0.29730709  0.02791478 -10.6505 < 2.2e-16 ***
## year1997         -0.28114768  0.02821249  -9.9654 < 2.2e-16 ***
## year1998         -0.29860441  0.02890728 -10.3297 < 2.2e-16 ***
## year1999         -0.30117699  0.03050502  -9.8730 < 2.2e-16 ***
## year2000         -0.32398300  0.02977715 -10.8803 < 2.2e-16 ***
## year2001         -0.31571197  0.02949408 -10.7042 < 2.2e-16 ***
## year2002         -0.29774671  0.03088602  -9.6402 < 2.2e-16 ***
## year2003         -0.30720913  0.03281396  -9.3621 < 2.2e-16 ***
## `event_time_-2`   0.00512458  0.03291328   0.1557 0.8763015
## `event_time_-3`   0.02953292  0.02902652   1.0174 0.3091890
## `event_time_-4`   0.03381551  0.03932954   0.8598 0.3901071
## `event_time_-5`   0.02253612  0.03072470   0.7335 0.4634356
## `event_time_-6`   0.04552640  0.03150866   1.4449 0.1488056
## `event_time_-7`   0.03830746  0.03185431   1.2026 0.2294243
## `event_time_-8`   0.01662750  0.03258818   0.5102 0.6100030
## `event_time_-9`   0.00167820  0.03880713   0.0432 0.9655152
## `event_time_-10`  0.01755795  0.03511047   0.5001 0.6171315
## `event_time_-11`  0.03051757  0.04546095   0.6713 0.5021906
## `event_time_-12`  0.02310302  0.04311881   0.5358 0.5922173
## `event_time_-13`  0.04820716  0.03406225   1.4153 0.1573040
## `event_time_-14`  0.01835077  0.03494207   0.5252 0.5995774
## `event_time_-15`  0.05965779  0.04202904   1.4194 0.1560840
## `event_time_-16`  0.06134474  0.05444834   1.1267 0.2601586
## `event_time_-17`  0.05307832  0.04570573   1.1613 0.2457968
## `event_time_-18` -0.00785960  0.08089817  -0.0972 0.9226235
## `event_time_-19`  0.01615047  0.08143728   0.1983 0.8428369
## `event_time_-20`  0.07670693  0.07747897   0.9900 0.3223980
## `event_time_-21`  0.12421231  0.06068096   2.0470 0.0409235 *
## `event_time_-22` -0.15448989  0.04417670  -3.4971 0.0004912 ***
## event_time_0     -0.03060715  0.03040689  -1.0066 0.3143790
```

```
## event_time_1     -0.04452815  0.03097459  -1.4376 0.1508710
## event_time_2     -0.04655569  0.02805608  -1.6594 0.0973551 .
## event_time_3     -0.04634076  0.02755122  -1.6820 0.0928860 .
## event_time_4     -0.06512714  0.03490050  -1.8661 0.0623248 .
## event_time_5     -0.08029593  0.03786273  -2.1207 0.0341935 *
## event_time_6     -0.09647172  0.03848419  -2.5068 0.0123422 *
## event_time_7     -0.07564843  0.03392623  -2.2298 0.0259842 *
## event_time_8     -0.13173038  0.04538268  -2.9027 0.0037819 **
## event_time_9     -0.11399294  0.04324213  -2.6362 0.0085160 **
## event_time_10    -0.11572393  0.03343698  -3.4610 0.0005612 ***
## event_time_11    -0.13358278  0.04530390  -2.9486 0.0032670 **
## event_time_12    -0.14867622  0.03481568  -4.2704 2.139e-05 ***
## event_time_13    -0.14076139  0.04244721  -3.3162 0.0009457 ***
## event_time_14    -0.13552964  0.04401710  -3.0790 0.0021339 **
## event_time_15    -0.14252426  0.03580109  -3.9810 7.362e-05 ***
## event_time_16    -0.19560653  0.03696740  -5.2913 1.494e-07 ***
## event_time_17    -0.20873954  0.04475258  -4.6643 3.520e-06 ***
## event_time_18    -0.27853562  0.03348964  -8.3171 2.949e-16 ***
## event_time_19    -0.29294767  0.03658735  -8.0068 3.276e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2

We now apply the synthetic control methods from Abadie et al (2010).

(a)

i. Compare the average pre-period log traffic fatalities per capita of the TU site to that of the average of all the "control" states. Next, graph the pre-period log traffic fatalities by year for the pre-period for both the TU and the average of the control group. Interpret.

```
# #Create a treatment status variable that = 1 if state is CT, IA, NM, or TX and =0 otherwise.
# traffic[,treat := ifelse(state_name == "CT" | state_name == "IA"|state_name == "NM" | state_name == "TX" |s
#
# traffic_TU <- traffic[treat == 0 | state == 99,]
# #Change treatment variable to a factor variable
# traffic_TU$treat <- as.factor(traffic$treat)
#
# #Create log fatalities per capita variable
# traffic_TU[, ln_fat_pc := log(fatalities/population)]
#
# #Compare the average pre-period log traffic fatalities per capita between treatment and control
# premeanT <- mean(traffic_TU[treat == 1 & year<1986, ln_fat_pc]) #mean pre-period log traffic fatalities in
# premeanC <- mean(traffic_TU[treat == 0 & year<1986, ln_fat_pc]) #mean pre-period log traffic fatalities in
#
# #Create variable of mean log traffic fatalities by treatment status by year
# traffic_TU[, mean_lnfat_treat := lapply(.SD, mean), .SDcols = c("ln_fat_pc"), by = c("treat","year")]
#
# #Graph the mean pre-period log traffic fatalities by year for Treatment vs Control
# traffic_TU[year < 1986,] %>%
#   ggplot(aes(x=year, y = mean_lnfat_treat, group = treat, color = treat)) +
#   geom_line() +
#   theme_minimal() +
#   labs(title = "Average Pre-Period Log Traffic Fatalities by Year", x = "Year", y = "Log Traffic Fatalities
#   theme(plot.title = element_text(hjust = 0.5)) +
#   scale_color_manual(labels = c("Controls", "TU"), values = c("coral1", "cyan3"))
```

The average pre-period log traffic fatalities per capita in our aggregate treatment unit is premean t compared with premean c in our control states. Graphically, we can see that while log traffic fatalities per capita are declining over time in both groups, treatment units have on average higher traffic fatalities per capita than control units in all pre-period years. This makes sense since states with higher traffic fatalities may make reducing traffic fatalities a bigger policy priority, leading to a higher likelihood of implementing seat belt laws.

```
#generate a variable that is the absolute value of the difference between the dependent variable of the TU si
# TU_dep_1985 <- traffic_TU[state == 99 & year == 1985, ln_fat_pc]
# traffic_TU[year == 1985,compare_dep := abs(ln_fat_pc - TU_dep_1985)]
# View(traffic_TU[compare_dep == min(traffic_TU$compare_dep),.(state, state_name)])
# apply(traffic_TU[year == 1985,ln_fat_pc])
```