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 $[1981, 2003]$.

The latest year that any state adopted a primary seat belt law was 2003, so the minimum event time value we can estimate in this dataset is -22. We know this because year 2003 is coded as event time 0 in this case, and there are 22 periods in $[1981, 2002]$ which come before the year of adoption so they are coded with negative event time values.

The earliest year that any state adopted a primary seat belt law was 1984, so the maximum event time value we can estimate in this dataset is 19. We know this because year 1984 is coded as event time 0 in this case, and there are 19 periods in $[1985, 2003]$ which come after the year of adoption so they are coded with positive event time values.

We have a single treated state ($s = 1$ if a state ever gets a primary seat belt law in the sample) and a single control state ($s = 0$ if a state never gets a primary seat belt law in the sample). 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."

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

  # find the row # and year corresponding to the first occurrence of primary == 1
  # in the dataframe for just that state
  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)
  # make all event times very negative (-1000) for these never treated states

  if(is.na(row_of_first_occ)){

    # all event_time values get coded as -1000
    setDT(traffic)[, event_time := ifelse(state == s, -1000, event_time)]
  }

  # case where a state was treated in the sample (row_of_first_occ > 1)
  # there are 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 time 2, etc.
      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 event time -2, etc.
      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
# multipy 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_cols]

# 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)]

# event study regression using all the event time indicators
es1 <- lm(ln_fat_pc ~ .-`event_time_-1000`, data = traf_reg)
es1_cluster <- vcovCL(es1, type = "HC1", cluster=traf_reg$county) # get cluster cov-var
```

```r
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
```

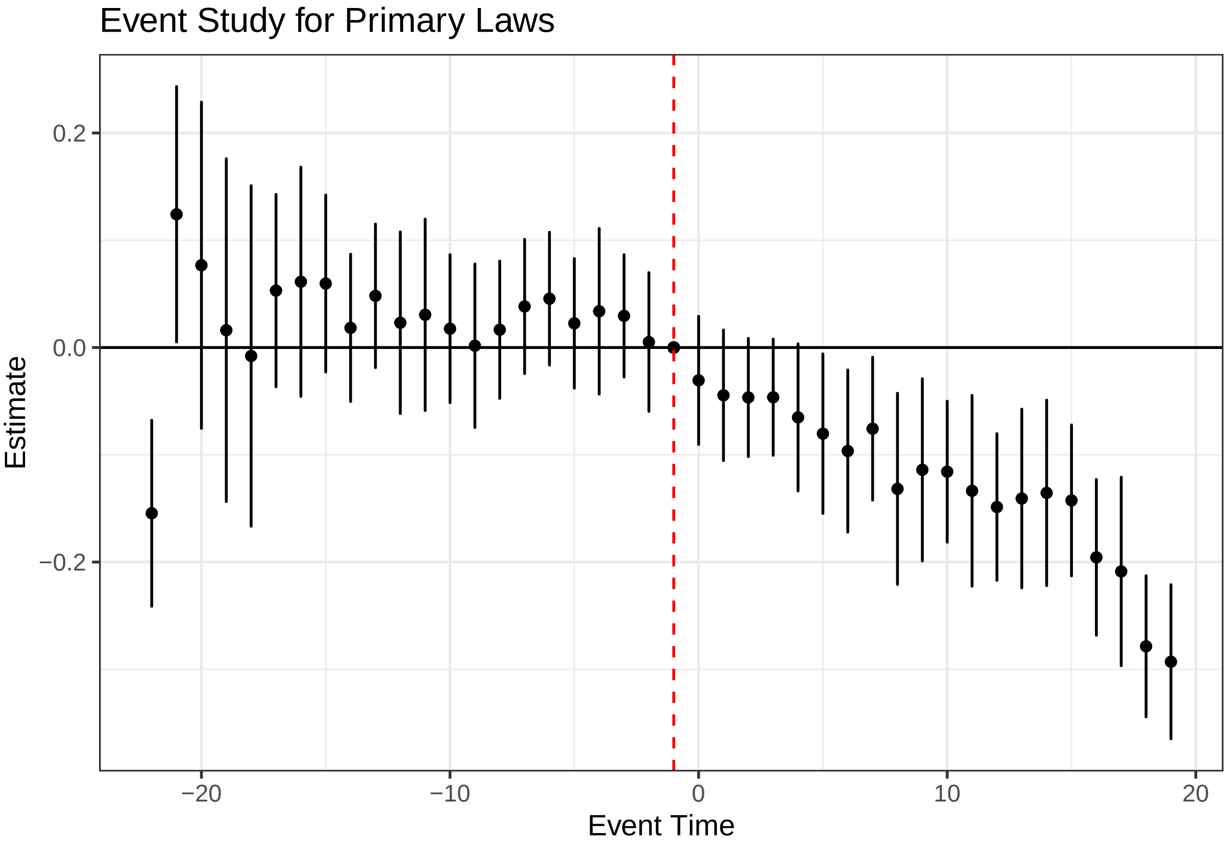
When we estimate an event study regression using all the event time indicators, the `lm` function in R chooses to remove event time 19. It makes sense that R omitted one event time indicator becuase in practice, one cannot estimate the regression using all the event time indicators because of the dummy variable trap. In an event study regression using all the event time indicators, 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 (the D_{jst}) 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_-1000`, 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
```

```r
es2_table <- data.table(tidy(coeftest(es2, vcov. = es2_cluster)))

es2_table <- es2_table[grep("event_time", term),]
es2_table[, event_time := as.numeric(str_match(es2_table$term, "([0-9]{1,2})")[, 2])]
es2_table[grep("-", term), event_time := -1 * event_time]
es2_table[, ci_upper := estimate + 1.96*std.error]
es2_table[, ci_lower := estimate - 1.96*std.error]

ggplot(data=es2_table, aes(x=event_time, y=estimate)) +
  geom_errorbar(aes(ymin=ci_lower, ymax=ci_upper), width=.1)  +
  geom_point() +
  geom_point(aes(x=-1, y=0), colour="black") +
  geom_hline(yintercept=0, colour="black") +
  geom_vline(xintercept = -1, colour = "red", linetype = "dashed") +
  xlab("Event Time") +
  ylab("Estimate") +
  ggtitle("Event Study for Primary Laws") +
  theme_bw()
```



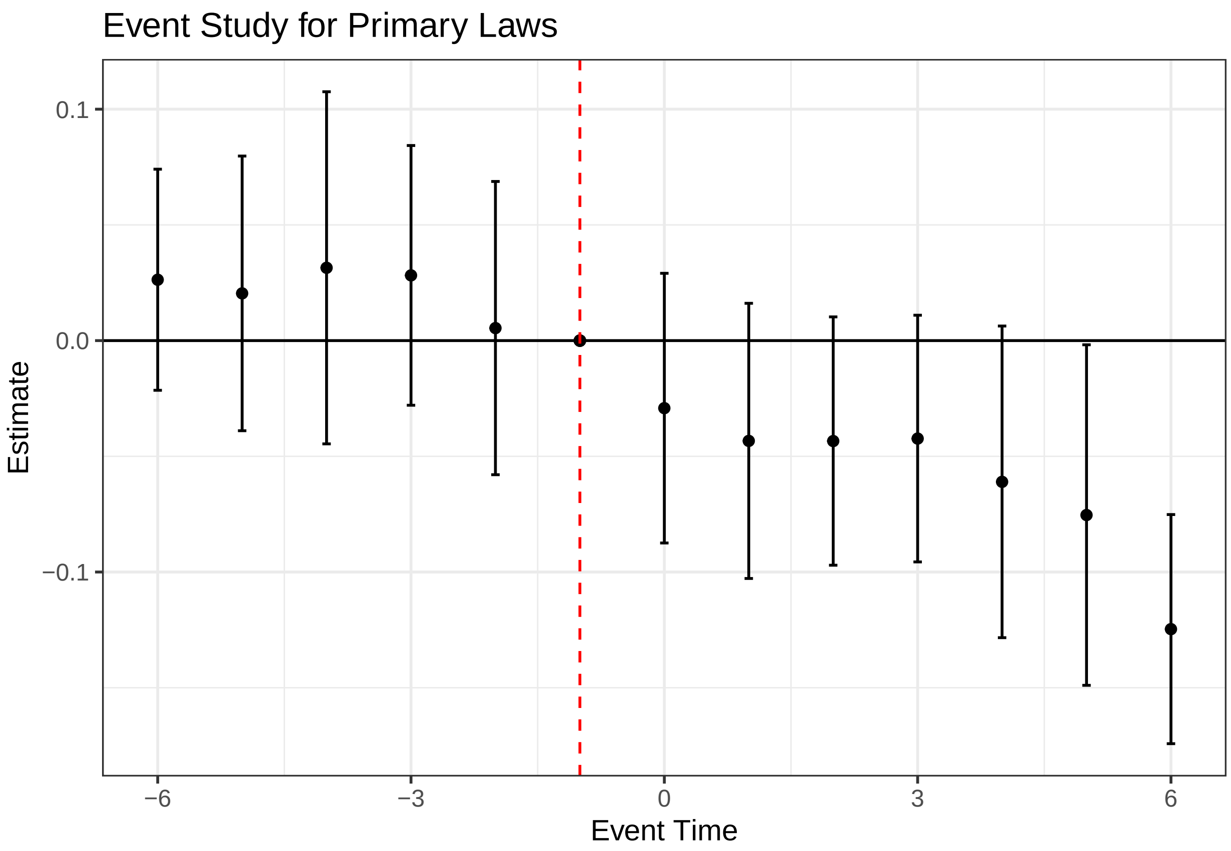
(c) Create minimum and maximum event time indicators that correspond to bins of event time < -5 and event time > 5 respectively. Appropriately specify and estimate an event study regression using these min and max event time indicators. Generate a plot of the event study coefficients. Explain which specification you prefer, this one or the one in part (b).

```
# create min and max event time indicators for event time < -5 and > 5
traf_pt1 <- traf_pt1[, bin_event_time := event_time]

# make any event time > 5 equal to 6 (max event time indicator)
setDT(traf_pt1)[, bin_event_time := ifelse(event_time > 5, 6, bin_event_time)]

# make any event time < -5 equal to -6 (min event time indicator)
# this means never treated states (which have event time -1000) will be in the
# min event time group, but this doesn't change the coefficient on the min
# event time dummy
setDT(traf_pt1)[, bin_event_time := ifelse(event_time < -5, -6, bin_event_time)]


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

# interact with ever treated
bin_event_time_cols <- names(traf_pt1[,`bin_event_time_-1`:`bin_event_time_6`]) # get list of bin event time
# multiply ever treated by all bin event time columns
traf_pt1 <- traf_pt1[, (bin_event_time_cols) := lapply(.SD, function(x) x * traf_pt1$treat),
                     .SDcols = bin_event_time_cols]

# make table with just the variables for the regression (makes writing out regression equation easier)
traf_reg3 <- traf_pt1[,c("ln_fat_pc", "state", "year",..bin_event_time_cols)]
traf_reg3[,state:=factor(state)]
```

```r
traf_reg3[,year:=factor(year)]

# event study regression using all the bin event time indicators except -1
es3 <- lm(ln_fat_pc ~ .-`bin_event_time_-1`, data = traf_reg3)
es3_cluster <- vcovCL(es3, type = "HC1", cluster=traf_reg3$county) # get cluster cov-var
coeftest(es3, vcov. = es3_cluster)
```

```
##
## t test of coefficients:
##
##                        Estimate Std. Error  t value  Pr(>|t|)    
## (Intercept)         -1.1851031  0.0429844 -27.5706 < 2.2e-16 ***
## state2              -0.0014786  0.0394365  -0.0375 0.9700990    
## state3              -0.0537635  0.0373840  -1.4381 0.1506998    
## state4              -0.4713032  0.0340104 -13.8576 < 2.2e-16 ***
## state5              -0.3678928  0.0386491  -9.5188 < 2.2e-16 ***
## state6              -0.7019412  0.0344632 -20.3678 < 2.2e-16 ***
## state7              -0.3386978  0.0367720  -9.2107 < 2.2e-16 ***
## state8              -0.1478758  0.0368406  -4.0139 6.408e-05 ***
## state9              -0.1185312  0.0283395  -4.1825 3.130e-05 ***
## state10             -0.3103641  0.0400031  -7.7585 2.075e-14 ***
## state11             -0.0447528  0.0353092  -1.2675 0.2052815    
## state12             -0.6346284  0.0355413 -17.8561 < 2.2e-16 ***
## state13             -0.3748021  0.0288828 -12.9766 < 2.2e-16 ***
## state14             -0.2755749  0.0379446  -7.2625 7.520e-13 ***
## state15             -0.1241064  0.0397942  -3.1187 0.0018673 ** 
## state16             -0.1386447  0.0354076  -3.9157 9.613e-05 ***
## state17             -1.0022098  0.0497059 -20.1628 < 2.2e-16 ***
## state18             -0.5708763  0.0289478 -19.7209 < 2.2e-16 ***
## state19             -0.3956733  0.0383985 -10.3044 < 2.2e-16 ***
## state20             -0.4813061  0.0287953 -16.7147 < 2.2e-16 ***
## state21             -0.6109083  0.0372360 -16.4064 < 2.2e-16 ***
## state22             -0.1863257  0.0413539  -4.5056 7.381e-06 ***
## state23              0.1876688  0.0457687   4.1004 4.452e-05 ***
## state24              0.1005756  0.0425361   2.3645 0.0182417 *  
## state25             -0.0887259  0.0336750  -2.6348 0.0085468 ** 
## state26             -0.4608823  0.0473997  -9.7233 < 2.2e-16 ***
## state27             -0.3525430  0.0398605  -8.8444 < 2.2e-16 ***
## state28             -0.6144820  0.0506305 -12.1366 < 2.2e-16 ***
## state29             -0.8303668  0.0316447 -26.2403 < 2.2e-16 ***
## state30              0.2902553  0.0326813   8.8814 < 2.2e-16 ***
## state31             -0.0980731  0.0463995  -2.1137 0.0347856 *  
## state32             -0.7913905  0.0348193 -22.7285 < 2.2e-16 ***
## state33             -0.5680936  0.0357867 -15.8744 < 2.2e-16 ***
## state34             -0.1112199  0.0394648  -2.8182 0.0049225 ** 
## state35             -0.2775531  0.0325362  -8.5306 < 2.2e-16 ***
## state36             -0.5729382  0.0350908 -16.3273 < 2.2e-16 ***
## state37             -1.0229572  0.0460235 -22.2268 < 2.2e-16 ***
## state38              0.0641215  0.0387942   1.6529 0.0986663 .  
## state39             -0.1184802  0.0451882  -2.6219 0.0088734 ** 
## state40             -0.0582602  0.0374543  -1.5555 0.1201367    
## state41             -0.1454963  0.0335610  -4.3353 1.600e-05 ***
## state42             -0.3800035  0.0384296  -9.8883 < 2.2e-16 ***
## state43             -0.4880906  0.0366336 -13.3236 < 2.2e-16 ***
## state44             -0.3403037  0.0461326  -7.3767 3.353e-13 ***
## state45             -0.5951598  0.0350879 -16.9620 < 2.2e-16 ***
## state46             -0.4703752  0.0356584 -13.1911 < 2.2e-16 ***
## state47             -0.0803987  0.0399798  -2.0110 0.0445894 *  
## state48              0.2677424  0.0483204   5.5410 3.826e-08 ***
## year1982            -0.1219330  0.0302597  -4.0296 6.003e-05 ***
## year1983            -0.1580702  0.0273465  -5.7803 9.899e-09 ***
## year1984            -0.1480845  0.0278130  -5.3243 1.246e-07 ***
## year1985            -0.1634447  0.0288481  -5.6657 1.903e-08 ***
## year1986            -0.1144578  0.0268552  -4.2620 2.213e-05 ***
## year1987            -0.1140221  0.0280642  -4.0629 5.218e-05 ***
## year1988            -0.1033062  0.0278800  -3.7054 0.0002224 ***
## year1989            -0.1623831  0.0275752  -5.8887 5.275e-09 ***
## year1990            -0.1816205  0.0257360  -7.0571 3.134e-12 ***
## year1991            -0.2532346  0.0249786 -10.1381 < 2.2e-16 ***
## year1992            -0.3067247  0.0254179 -12.0673 < 2.2e-16 ***
## year1993            -0.3049839  0.0247450 -12.3251 < 2.2e-16 ***
## year1994            -0.3101508  0.0260842 -11.8904 < 2.2e-16 ***
## year1995            -0.2853803  0.0265206 -10.7607 < 2.2e-16 ***
## year1996            -0.2973632  0.0263709 -11.2762 < 2.2e-16 ***
## year1997            -0.2821159  0.0265267 -10.6352 < 2.2e-16 ***
## year1998            -0.3013011  0.0273598 -11.0125 < 2.2e-16 ***
## year1999            -0.3041721  0.0288673 -10.5369 < 2.2e-16 ***
## year2000            -0.3269499  0.0279770 -11.6864 < 2.2e-16 ***
## year2001            -0.3207320  0.0276072 -11.6177 < 2.2e-16 ***
## year2002            -0.3075644  0.0289432 -10.6265 < 2.2e-16 ***
## year2003            -0.3191398  0.0306387 -10.4162 < 2.2e-16 ***
## `bin_event_time_-2`  0.0054120  0.0323290   0.1674 0.8670860    
## `bin_event_time_-3`  0.0281883  0.0286295   0.9846 0.3250592    
## `bin_event_time_-4`  0.0314695  0.0388176   0.8107 0.4177255    
## `bin_event_time_-5`  0.0204130  0.0302775   0.6742 0.5003386    
## `bin_event_time_-6`  0.0263056  0.0243718   1.0793 0.2806887    
## bin_event_time_0    -0.0291829  0.0297264  -0.9817 0.3264718    
## bin_event_time_1    -0.0433348  0.0303345  -1.4286 0.1534355    
## bin_event_time_2    -0.0434003  0.0273779  -1.5852 0.1132238    
## bin_event_time_3    -0.0423315  0.0271933  -1.5567 0.1198536    
## bin_event_time_4    -0.0610308  0.0343556  -1.7764 0.0759575 .  
## bin_event_time_5    -0.0753784  0.0375447  -2.0077 0.0449381 *  
## bin_event_time_6    -0.1246710  0.0252594  -4.9356 9.324e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
es3_table <- data.table(tidy(coeftest(es3, vcov. = es3_cluster)))

es3_table <- es3_table[grep("event_time", term),]
es3_table[, event_time := as.numeric(str_match(es3_table$term, "([0-9]{1,2})")[, 2])]
es3_table[grep("-", term), event_time := -1 * event_time]
es3_table[, ci_upper := estimate + 1.96*std.error]
es3_table[, ci_lower := estimate - 1.96*std.error]

ggplot(data=es3_table, aes(x=event_time, y=estimate)) +
  geom_errorbar(aes(ymin=ci_lower, ymax=ci_upper), width=.1)  +
  geom_point() +
  geom_point(aes(x=-1, y=0), colour="black") +
  geom_hline(yintercept=0, colour="black") +
  geom_vline(xintercept = -1, colour = "red", linetype = "dashed") +
  xlab("Event Time") +
  ylab("Estimate") +
  ggtitle("Event Study for Primary Laws") +
  theme_bw()
```



Explain which specification you prefer

(d) What happens to your estimates from part (b) if you exclude the "pure control" states from your sample? What about if you exclude the pure controls in part (c)?

```
# exclude "pure control" states from the sample
traf_pt1_no_pure <- traf_pt1[!(event_time==-1000),]

# redo part (b) with no "pure control" states

# make table with just the variables for the regression (makes writing out regression equation easier)
traf_reg <- traf_pt1_no_pure[,c("ln_fat_pc", "state", "year",..event_time_cols)]
traf_reg[,state:=factor(state)]
traf_reg[,year:=factor(year)]

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

```
##
## t test of coefficients:
##
##                      Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)       -0.93867626  0.06710440 -13.9883 < 2.2e-16 ***
## state4            -0.57113151  0.03981089 -14.3461 < 2.2e-16 ***
## state6            -0.90117911  0.04284820 -21.0319 < 2.2e-16 ***
## state7            -0.29596201  0.03806431  -7.7753 9.422e-14 ***
## state9            -0.17742868  0.03057832  -5.8024 1.518e-08 ***
## state10           -0.50960202  0.04713303 -10.8120 < 2.2e-16 ***
## state13           -0.40449993  0.03122113 -12.9560 < 2.2e-16 ***
## state16          -0.19754211  0.03777923  -5.2289 3.015e-07 ***
## state18          -0.60057416  0.03083970 -19.4741 < 2.2e-16 ***
## state20          -0.48130606  0.03091525 -15.5686 < 2.2e-16 ***
## state25          -0.27725245  0.04138925  -6.6987 8.938e-11 ***
## state29          -0.83036675  0.03443848 -24.1116 < 2.2e-16 ***
## state30           0.09101734  0.04116036   2.2113 0.0276929 *
## state32          -1.00756212  0.03918182 -25.7150 < 2.2e-16 ***
## state34          -0.14091769  0.04142506  -3.4017 0.0007508 ***
## state35          -0.40616623  0.03716714 -10.9281 < 2.2e-16 ***
## state41          -0.34473428  0.04348789  -7.9271 3.391e-14 ***
## state45          -0.57389342  0.03720712 -15.4243 < 2.2e-16 ***
## year1982         -0.11620968  0.04824693  -2.4086 0.0165529 *
## year1983         -0.20051293  0.03868325  -5.1835 3.779e-07 ***
## year1984         -0.17817067  0.04437018  -4.0155 7.329e-05 ***
## year1985         -0.21255244  0.04104450  -5.1786 3.872e-07 ***
## year1986         -0.16733758  0.03985644  -4.1985 3.450e-05 ***
## year1987         -0.17330300  0.04547929  -3.8106 0.0001650 ***
## year1988         -0.16162521  0.04270246  -3.7849 0.0001822 ***
## year1989         -0.23989972  0.04138687  -5.7965 1.568e-08 ***
## year1990         -0.27076157  0.04121944  -6.5688 1.946e-10 ***
## year1991         -0.38021678  0.04054914  -9.3767 < 2.2e-16 ***
## year1992         -0.44288796  0.04394472 -10.0783 < 2.2e-16 ***
## year1993         -0.45521041  0.03920320 -11.6116 < 2.2e-16 ***
## year1994         -0.46672318  0.04296148 -10.8638 < 2.2e-16 ***
## year1995         -0.45800622  0.04625382  -9.9020 < 2.2e-16 ***
## year1996         -0.46715047  0.04566603 -10.2297 < 2.2e-16 ***
## year1997         -0.48545779  0.05203253  -9.3299 < 2.2e-16 ***
## year1998         -0.53783753  0.05176878 -10.3892 < 2.2e-16 ***
## year1999         -0.58064643  0.06135309  -9.4640 < 2.2e-16 ***
## year2000         -0.60503659  0.05398769 -11.2069 < 2.2e-16 ***
## year2001         -0.57855271  0.05182837 -11.1629 < 2.2e-16 ***
## year2002         -0.61069031  0.05690772 -10.7312 < 2.2e-16 ***
## year2003         -0.61068375  0.04209888 -14.5059 < 2.2e-16 ***
## `event_time_-2`  -0.01827990  0.03563161  -0.5130 0.6082732
## `event_time_-3`  -0.00139343  0.03232592  -0.0431 0.9656432
## `event_time_-4`  -0.02215910  0.04317341  -0.5133 0.6081102
## `event_time_-5`  -0.04220906  0.03720777  -1.1344 0.2574339
## `event_time_-6`  -0.04421070  0.03817433  -1.1581 0.2476399
## `event_time_-7`  -0.06434532  0.04226425  -1.5225 0.1288419
## `event_time_-8`  -0.10280128  0.04327928  -2.3753 0.0180987 *
## `event_time_-9`  -0.12964684  0.05111401  -2.5364 0.0116540 *
## `event_time_-10` -0.12948594  0.04672029  -2.7715 0.0058923 **
## `event_time_-11` -0.12946178  0.05756914  -2.2488 0.0251762 *
## `event_time_-12` -0.14958645  0.05619820  -2.6618 0.0081490 **
## `event_time_-13` -0.13285875  0.04803180  -2.7661 0.0059895 **
## `event_time_-14` -0.18084119  0.05348488  -3.3812 0.0008074 ***
## `event_time_-15` -0.14011990  0.05744322  -2.4393 0.0152368 *
## `event_time_-16` -0.16388497  0.07070782  -2.3178 0.0210653 *
## `event_time_-17` -0.16808063  0.06274546  -2.6788 0.0077550 **
## `event_time_-18` -0.26056236  0.09521369  -2.7366 0.0065397 **
## `event_time_-19` -0.23384180  0.09638790  -2.4260 0.0157933 *
## `event_time_-20` -0.19253753  0.08681456  -2.2178 0.0272408 *
## `event_time_-21` -0.16413950  0.08839230  -1.8569 0.0641994 .
## `event_time_-22` -0.44602858  0.06938238  -6.4286 4.449e-10 ***
## event_time_0     -0.01843273  0.03202272  -0.5756 0.5652636
## event_time_1     -0.02187265  0.03360617  -0.6509 0.5155893
## event_time_2     -0.00606362  0.02917824  -0.2078 0.8355016
## event_time_3      0.00017107  0.02909842   0.0059 0.9953126
## event_time_4      0.00180118  0.03683352   0.0489 0.9610278
## event_time_5     -0.00217556  0.03522155  -0.0618 0.9507847
## event_time_6      0.00278125  0.03764740   0.0739 0.9411530
## event_time_7      0.03489121  0.03138791   1.1116 0.2671040
## event_time_8     -0.01039932  0.04310786  -0.2412 0.8095175
## event_time_9      0.02312210  0.03910720   0.5912 0.5547535
## event_time_10     0.02193672  0.02909652   0.7539 0.4514228
## event_time_11     0.03612245  0.04666453   0.7741 0.4394263
## event_time_12     0.04492761  0.03829412   1.1732 0.2415417
## event_time_13     0.08743980  0.05005985   1.7467 0.0816077 .
## event_time_14     0.09651267  0.04906039   1.9672 0.0499849 *
## event_time_15     0.09243268  0.03717605   2.4863 0.0133943 *
## event_time_16     0.07143818  0.03822860   1.8687 0.0625383 .
## event_time_17     0.05117408  0.03712668   1.3784 0.1690135
## event_time_18     0.02388103  0.03450166   0.6922 0.4893112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# redo part (c) with no "pure control" states

# make table with just the variables for the regression (makes writing out regression equation easier)
traf_reg3 <- traf_pt1_no_pure[,c("ln_fat_pc", "state", "year",..bin_event_time_cols)]
traf_reg3[,state:=factor(state)]
traf_reg3[,year:=factor(year)]

# event study regression using all the bin event time indicators except -1
es_c <- lm(ln_fat_pc ~ .-`bin_event_time_-1`, data = traf_reg3)
es_c_cluster <- vcovCL(es_c, type = "HC1", cluster=traf_reg3$county) # get cluster cov-var
coeftest(es_c, vcov. = es_c_cluster)
```

```
##
## t test of coefficients:
##
##                       Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)        -1.12994343  0.06286414 -17.9744 < 2.2e-16 ***
## state4             -0.49846956  0.04074987 -12.2324 < 2.2e-16 ***
## state6             -0.75901700  0.04518098 -16.7995 < 2.2e-16 ***
## state7             -0.33129392  0.03782764  -8.7580 < 2.2e-16 ***
## state9             -0.13287921  0.03197190  -4.1561 4.044e-05 ***
## state10            -0.36743991  0.04918957  -7.4699 6.070e-13 ***
## state13            -0.38060451  0.03191764 -11.9246 < 2.2e-16 ***
## state16            -0.15299264  0.03887009  -3.9360 9.936e-05 ***
## state18            -0.57667874  0.03186720 -18.0963 < 2.2e-16 ***
## state20            -0.48130606  0.03097098 -15.5406 < 2.2e-16 ***
## state25            -0.14152891  0.04331994  -3.2671 0.0011906 **
## state29            -0.83036675  0.03292296 -25.2215 < 2.2e-16 ***
## state30             0.23317945  0.04363247   5.3442 1.611e-07 ***
## state32            -0.85446696  0.04512494 -18.9356 < 2.2e-16 ***
## state34            -0.11702227  0.04221294  -2.7722 0.0058559 **
## state35            -0.31326500  0.03854476  -8.1273 7.030e-15 ***
## state41            -0.20257217  0.04537283  -4.4646 1.073e-05 ***
## state45            -0.59002058  0.03518402 -16.7696 < 2.2e-16 ***
## year1982           -0.10804607  0.04871607  -2.2179 0.0271837 *
## year1983           -0.16566444  0.03808948  -4.3493 1.778e-05 ***
## year1984           -0.15077682  0.04058665  -3.7149 0.0002353 ***
## year1985           -0.16043616  0.04009355  -4.0015 7.634e-05 ***
## year1986           -0.11587276  0.03967392  -2.9206 0.0037122 **
## year1987           -0.10372162  0.04628237  -2.2411 0.0256278 *
## year1988           -0.09011181  0.04269242  -2.1107 0.0354816 *
## year1989            -0.15711002  0.04007061  -3.9208 0.0001056 ***
## year1990            -0.17314079  0.03950413  -4.3829 1.537e-05 ***
## year1991            -0.27392692  0.04024700  -6.8061 4.173e-11 ***
## year1992            -0.32169005  0.04156648  -7.7392 1.008e-13 ***
## year1993            -0.31861389  0.03954330  -8.0573 1.144e-14 ***
## year1994            -0.33033795  0.04234940  -7.8003 6.667e-14 ***
## year1995            -0.31084071  0.04403631  -7.0587 8.625e-12 ***
## year1996            -0.31074550  0.04531481  -6.8575 3.039e-11 ***
## year1997            -0.31545404  0.04955590  -6.3656 5.892e-10 ***
## year1998            -0.36044617  0.05064228  -7.1175 5.941e-12 ***
## year1999            -0.38796962  0.05427160  -7.1487 4.871e-12 ***
## year2000            -0.40067650  0.05115997  -7.8318 5.382e-14 ***
## year2001            -0.37316224  0.05060426  -7.3741 1.137e-12 ***
## year2002            -0.40159795  0.05210111  -7.7080 1.243e-13 ***
## year2003            -0.40125169  0.05379953  -7.4583 6.551e-13 ***
## `bin_event_time_-2` -0.00341280  0.03302124  -0.1034 0.9177412
## `bin_event_time_-3`  0.01820618  0.02924025   0.6226 0.5339124
## `bin_event_time_-4`  0.01087455  0.04008606   0.2713 0.7863303
## `bin_event_time_-5`  0.00072493  0.03303626   0.0219 0.9825051
## `bin_event_time_-6` -0.02310317  0.03011315  -0.7672 0.4434555
## bin_event_time_0    -0.02464701  0.03127419  -0.7881 0.4311573
## bin_event_time_1    -0.04065655  0.03289836  -1.2358 0.2173257
## bin_event_time_2    -0.03248556  0.02994641  -1.0848 0.2787367
## bin_event_time_3    -0.03386100  0.03112662  -1.0878 0.2773864
## bin_event_time_4    -0.04579845  0.03842360  -1.1919 0.2340674
## bin_event_time_5    -0.05597270  0.03987283  -1.4038 0.1612415
## bin_event_time_6    -0.07580574  0.03322343  -2.2817 0.0230873 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

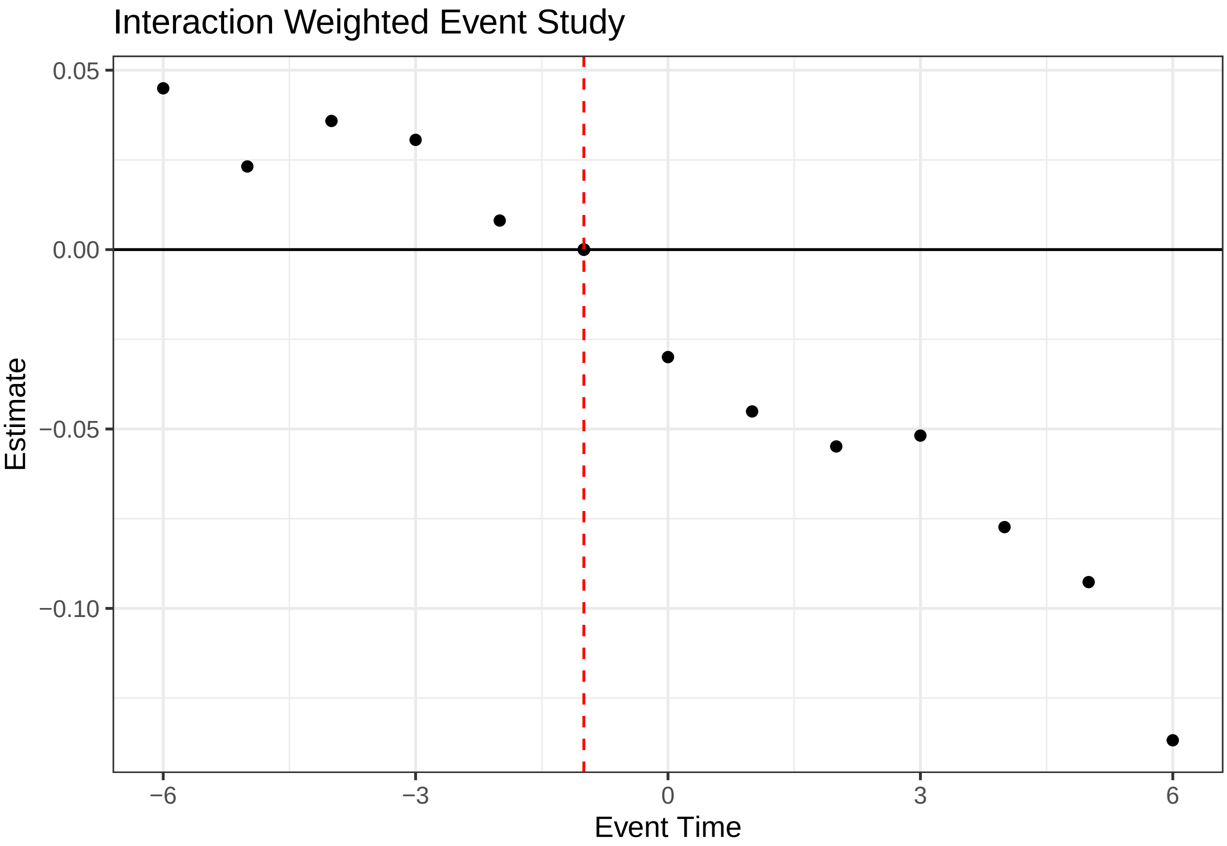
What happens + explanation

(e)

(f) Building off the event study regression from part (c), estimate the interaction weighted event study estimator from Sun and Abraham (2020) (discussed in lecture and posted on bCourses for refrence). As a reminder, the interacted event study regression takes the standard event time indicators (without any binning) and interacts each one with a cohort indicator (a cohort refers to a group of states that share the same date on which they were first treated). You then form the estimate for event time coefficient τ_j by averaging the estimates of the cohort-specific τ_j using the weights described in Sun and Abraham (2020).

```
## f interaction weighted
# get cohorts
traf_pt1[event_time==0,cohort:=year]
traf_pt1[,cohort:= lapply(.SD, function(x) replace(x, is.na(x), mean(x, na.rm=T))), .SDcols = "cohort", by =
traf_pt1[is.na(cohort), cohort:=2103]
# get dummy for each cohort
traf_pt1 <- traf_pt1[,dummy_cols(traf_pt1, select_columns = c("cohort"))]
cohort_cols <- names(traf_pt1[,cohort_1984:cohort_2003]) # get list of bin event time columns

traf_reg_iw <- traf_pt1[,c("ln_fat_pc", "state", "event_time", "cohort", "year",..event_time_cols,..cohort_co
traf_reg_iw[,state:=factor(state)]
traf_reg_iw[,year:=factor(year)]

# get interactions
reg_iw_interactions <- function(x) {

  dt <- traf_reg_iw[cohort==x,]
  min_et <- 1981 - x # min interaction
```

```r
  max_et <- 2003 - x # max interaction

  # don't create interactions for never treated
  if (x != 2103) {
    # loop through all in between min and max
    for (i in min_et:max_et) {
      dt[, glue("int_{x}_{i}") := eval(as.name(glue("event_time_{i}"))) * eval(as.name(glue("cohort_{x}")))]
    }
  }

  else {
  }

  return(dt)
}

cohorts <- unique(traf_reg_iw$cohort) # unique cohorts
int_dt_list <- lapply(cohorts, reg_iw_interactions) # lapply through all cohorts
int_dt <- rbindlist(int_dt_list, fill = T) # get as table
int_cols <- names(int_dt[,grep("int", names(int_dt)), with = F])
int_dt[,(int_cols):= lapply(.SD, function(x) replace(x, is.na(x), 0)), .SDcols = int_cols]
traf_reg_iw_final <-  int_dt[,c("ln_fat_pc", "state", "year",..int_cols)]

# run regression. omitted cohort is never treated. omitted event time is -1
es_iw <- lm(ln_fat_pc ~. - `int_2000_-1` - `int_1993_-1` - `int_1986_-1` - `int_2003_-1` -
       `int_1996_-1` - `int_1998_-1` - `int_1987_-1` - `int_1984_-1` - `int_1991_-1` - `int_2002_-1`,
      data = traf_reg_iw_final)

# get results
iw_tab <- data.table(tidy(coeftest(es_iw)))
iw_tab <- iw_tab[grep("int", term),]
iw_tab[, cohort := as.numeric(str_extract(iw_tab$term, "([0-9]{4})"))]
iw_tab[, event_time := -1*as.numeric(str_match(iw_tab$term, "_-([0-9]{1,2})")[,2])]
iw_tab[is.na(event_time), event_time := as.numeric(str_match(iw_tab[is.na(event_time),term], "([0-9]{1,2})$")

# get number of states
state_dt <- traf_pt1[,.(cohort = mean(cohort)), by = state]
state_dt <- state_dt[,.(num_states = .N), by = cohort]

iw_tab_merge <- merge(iw_tab, state_dt, by = "cohort")
# weighted average across cohorts
iw_tab_weight1 <- iw_tab_merge[,.(estimate = weighted.mean(estimate, w = num_states),
                                 num_states = .N), by = event_time]
# weighted average for t < -5 and t > 5
iw_tab_weight1[,time_bin := event_time]
iw_tab_weight1[event_time < -5,time_bin := -6]
iw_tab_weight1[event_time > 5,time_bin := 6]

iw_tab_weight2 <- iw_tab_weight1[,.(estimate = weighted.mean(estimate, w = num_states)), by = time_bin]

# plot
ggplot(data=iw_tab_weight2, aes(x=time_bin, y=estimate)) +
  geom_point() +
  geom_point(aes(x=-1, y=0), colour="black") +
  geom_hline(yintercept=0, colour="black") +
  geom_vline(xintercept = -1, colour = "red", linetype = "dashed") +
  xlab("Event Time") +
  ylab("Estimate") +
  ggtitle("Interaction Weighted Event Study") +
  theme_bw()
```



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])
```