Event study design: How normalisation affects estimation and inference

Seiro Ito

2024年03月13日 16:48

Contents

Read marriage/divorce files

NCH marriage data is at state level but complete

Estimation

Identification strategy

Consider an event-study design for a unit i that starts being treated at τ:

$$y_{i,t} = a_t + a_i + \sum_{s=-L}^{G} \gamma_j D_{i,t+s} + \mathbf{b}'\mathbf{x}_{i,t} + e_{i,t},$$

where $D_{i,t+s} = 0, 1$ is an indicator function equalts to 1 if i is treated in $t + s$ with $s = -L, -L + 1, \cdots, -1, 0, 1, 2, \cdots, G-1, G$. Given i starts getting treated at τ, $D_{i,t-s} = 0$ for $t - s < \tau$, $D_{i,t-a} = 1$ for $t - s \geqslant \tau$. The indicator t measures the calendar time, s measures the event time (time-since-event). So we know that time-since-event s is equal to $t - \tau$, or $s = t - \tau$ or $t = s + \tau$. $\mathbf{x}_{it}$ is a vector of exogenous covariates.

Our data structure is the hybrid type based on the classification by Miller (2023):

- Treatment dates vary by units.
- There are never-treated units.

The key identifying assumptions are:

1. In the absence of treatments, all the units share the same time effects a_t (conditional on unit fixed effects a_i and covariates $\mathbf{x}_{i,t}$).
2. Selection of treatment timing, selection of treated or never-treated (by the end of our observation period) units are as good as random, given time and unit fixed effects and covariates.

The treatment effect parameters γ need to give differneces relative to a specific benchmark. It is common that to choose the benchmark of the mean pre-treatment effect, or setting $\gamma_{-1} = 0$.

To avoid multicollinearity, we need to drop one period FE from a_t, and one unit FE from a_i.

The number of pre-treatment periods need to balance efficiency and bias tradeoff. Longer perods provide efficiency, but it risks the inclusion of irrelevant periods, such as under marital market disruption immediately after the world war II (1946 onwards). Number of states reporting marriages (*Marriage Reporting Area*) increased from 32 (1957) to 37 (1964). Non MRA states also report data by using central files or survey estimation. In 1960, 33 MRA states, 8 states and DC hold central files of marriage records to construct data. In 1961-63, 35 MRA states 10 states with central files. In 1964, 37 MRA states, 7 states[*1] have central files.

46 states report number of marriages performed, 5 States and DC report the number of marriage licenses issued using central files. Texas only reports data for 10 counties.

To avoid the bias while not throwing away too much of efficiency, we choose 1961, 14 years before the landmark *Dunn v Palermo*, with 45 states in marriage data, as the starting year.

Miller (2023) recommends to base entire pre-period to be the reference period.

```
m2L <- qread(FPath("save", "m2L.qs"))
d12L <- qread(FPath("save", "d12L.qs"))
m2L <- m2L[!grepl("Cent|Mid|Mount|Eng|east|Pac|^South$|Atl?a|Unit|^West$", StateName), ]
d12L <- d12L[!grepl("Cent|Mid|Mount|Eng|east|Pac|^South$|Atl?a|Unit|^West$", StateName), ]
destat(m2L[, .(NumberOfStates=.N, NumberOfEntries=length(v[!is.na(v)])), by = time])
```

```
                  min 25\\% median 75\\%  max mean std 0s NAs  n
time             1956  1964   1972  1980 1988 1972 9.7  0    0 33
NumberOfStates     51    51     51    51   51   51 0.0  0    0 33
NumberOfEntries    51    51     51    51   51   51 0.0  0    0 33
```

Miller (2023) recommends to base entire pre-period to be the reference period. In the paper's accompanying code, he uses `cnsreg` of stata. This is to impose a linear restriction on the estimated parameters in OLS using minimization of the Lagrangian:

$$\mathcal{L} = SSE + \lambda[\bar{\gamma}_{pre}].$$

Stata's manual on `cnsreg` states that it uses a linear formula which should be similar to Hansen (2022), 8.8.[*2]

[*1]Why decreased?

[*2]Because this code is proprietary, one cannot see what it does.

However, in the current case, constrained least squares is not necessary. One can impose a set of nonzero constraints on γ_s for $s < 0$. Setting and substituting $\bar{\gamma}_{pre} = 0$ changes the estimating equation:

$$\bar{\gamma}_{pre} = 0 \quad \Leftrightarrow \quad \gamma_{-L} = -\sum_{s=-(L-1)}^{-1} \gamma_s, \tag{c}$$

so

$$\begin{aligned}
y_{i,t} &= a_t + a_i + \sum_{s=-L}^{G} \gamma_s D_{i,t+s} + \mathbf{b}'\mathbf{x}_{i,t} + e_{i,t}, \\
&= a_t + a_i - \left(\gamma_{-(L-1)} + \cdots + \gamma_{-1}\right) D_{i,t-L} + \gamma_{-(L-1)} D_{i,t-(L-1)} + \cdots + \gamma_{-1} D_{i,t-1} \\
&\quad + \gamma_0 D_{i,t} + \cdots + \gamma_G D_{i,t+G} + \mathbf{b}'\mathbf{x}_{i,t} + e_{i,t}, \\
&= a_t + a_i + \gamma_{-(L-1)} \left(D_{i,t-(L-1)} - D_{i,t-L}\right) + \cdots + \gamma_{-1} \left(D_{i,t-1} - D_{i,t-L}\right) \\
&\quad + \gamma_0 D_{i,t} + \cdots + \gamma_G D_{i,t+G} + \mathbf{b}'\mathbf{x}_{i,t} + e_{i,t}, \\
&= a_t + a_i + \sum_{s=-(L-1)}^{-1} \gamma_s \left(D_{i,t+s} - D_{i,t-L}\right) + \sum_{s=0}^{G} \gamma_s D_{i,t+s} + \mathbf{b}'\mathbf{x}_{i,t} + e_{i,t}.
\end{aligned}$$

Checking data problems

Click here to see data problem checks.

Anomalous entries.

```
d12L[abs(vs)> 3, ][order(StateName, time)]
```

```
              StateName time  case      vs    pop    v
1:             Colorado 1977 20557 3.07956 268420  7.7
2: District of Columbia 1979  4488 3.10579  65865  6.8
3: District of Columbia 1980  4682 3.39990  63975  7.3
4:             Illinois 1959 22700 3.25350 423648  5.4
5:               Nevada 1959  9509 4.02715  14004 67.9
```

```
m2L[abs(vs)> 3, ][order(time, StateName)]
```

```
       StateName time    v      vs
 1:      Arizona 1956 25.1 5.70298
 2:      Indiana 1956 16.5 3.97960
 3:  Mississippi 1956 31.2 4.31601
 4:   New Mexico 1956 26.2 4.66833
 5:      Georgia 1957 18.4 4.10033
 6:      Indiana 1957 16.5 3.97960
 7:  Mississippi 1957 29.3 3.93245
 8: Rhode Island 1957  8.9 3.18665
 9: Rhode Island 1959  8.8 3.03912
10:     Nebraska 1970 10.6 3.09488
11: Pennsylvania 1973  8.5 3.02942
12:         Iowa 1978  9.6 3.01838
13:         Iowa 1979  9.6 3.01838
14:     New York 1984  9.5 3.03035
15:     Kentucky 1988 13.3 3.07482
```

Data in 1956-1958 are unreliable that they use estimates. Drop from data.

```
d3L <- d12L[time >= 1959, ]
m3L <- m2L[time >= 1959, ]
```

```r
d3L[, vs := v/var(v)^(.5), by = .(StateName)]
d3L[, vs := vs-mean(vs[1961 <= time & time <= 1965]), by = .(StateName)]
m3L[, vs := v/var(v)^(.5), by = .(StateName)]
m3L[, vs := vs-mean(vs[1961 <= time & time <= 1965]), by = .(StateName)]
qsave(d3L, "../save/d3L.qs")
qsave(m3L,"../save/m3L.qs")
```

Anomalous entries.

```r
d3L[abs(vs)> 3, ][order(StateName, time)]
```

```
              StateName time  case      vs    pop    v
1:             Colorado 1977 20557 3.17652 268420  7.7
2: District of Columbia 1979  4488 3.14189  65865  6.8
3: District of Columbia 1980  4682 3.43942  63975  7.3
4:             Illinois 1959 22700 3.32992 423648  5.4
5:               Nevada 1959  9509 4.21927  14004 67.9
```

```r
d3L[vs < -.5, ][order(StateName, time)]
```

```
     StateName time  case        vs    pop    v
 1:    Arizona 1960  4780 -1.566809 130167  3.7
 2:     Hawaii 1966   897 -0.613358  71506  1.3
 3:    Indiana 1959  8228 -0.964599 452607  1.8
 4:     Nevada 1970  9138 -0.742205  48876 18.7
 5:     Nevada 1971  9474 -0.742205  50708 18.7
 6:     Nevada 1973  9975 -0.792626  54792 18.2
 7:     Nevada 1974 10045 -0.863216  57286 17.5
 8:     Nevada 1975 10542 -0.832964  59184 17.8
 9:     Nevada 1976 10298 -1.024565  64689 15.9
10:     Nevada 1977 10280 -1.095156  67782 15.2
11:     Nevada 1978 11213 -1.054818  71878 15.6
12:     Nevada 1979 11787 -1.074987  76525 15.4
13:     Nevada 1980 13842 -0.883385  80065 17.3
14:     Nevada 1981 14925 -0.853132  84600 17.6
15:     Nevada 1982 13092 -1.125408  87598 14.9
16:     Nevada 1983 13438 -1.115324  89726 15.0
17:     Nevada 1984 13822 -0.984228  84600 16.3
18:     Nevada 1985 13318 -1.095156  87598 15.2
19:     Nevada 1986 13470 -1.115324  89726 15.0
20:     Nevada 1987 13936 -1.085071  91078 15.3
21:     Nevada 1988 13922 -1.125408  93674 14.9
22: New Mexico 1967  1545 -0.827416 100402  1.5
23:       Utah 1959  1336 -1.035568  87455  1.5
     StateName time  case        vs    pop    v
```

```r
m3L[abs(vs)> 3, ][order(time, StateName)]
```

```
       StateName time    v       vs
 1: Rhode Island 1959  8.8  3.20044
 2:   New Mexico 1968  7.6 -3.48848
 3:     Nebraska 1970 10.6  3.06216
 4:      Arizona 1973 12.7  3.12131
 5: Pennsylvania 1973  8.5  3.09598
 6:     Virginia 1973 12.1  3.02976
 7:      Arizona 1974 12.6  3.06128
```

```
 8:      Wyoming 1974 16.8  3.14669
 9:     Delaware 1983  9.2  3.05812
10:     Arkansas 1988 14.6  3.39000
11:     Kentucky 1988 13.3  3.24121
```

Anomalous values of below can be dropped without much costs to data availability (year before 1960).

- Illinois, Nevada from divorce rate estimation.
- Rhode Island from marriage rate estimation.

However, it is easiest to set the starting year as 1960.

Stationarity tests.

```
library(tseries)
  d3L[, outcome := "divorce"]
  m3L[, outcome := "marriage"]
  dm3L <- rbind(d3L, m3L, use.names = T, fill = T)
      if (nrow(dm3L[is.na(v), ]) > 0)
    dm3L2 <- dm3L[!is.na(v), ] else
    dm3L2 <- dm3L
  stt <-  dm3L2[, .(
      kpss = kpss.test(v, null = "Trend")$p.value,
      adf = adf.test(v, alternative = "stationary", k = 5)$p.value),
      by = .(outcome, StateName)][kpss < .1 & adf < .1, ]
  print(
    sttW <- reshape(stt, direction = "wide", idvar = "StateName",
      timevar = "outcome", v.names = grepout("k|adf", colnames(stt)))
   )
```

```
        StateName kpss.divorce adf.divorce kpss.marriage adf.marriage
1:   North Dakota    0.0445779        0.01            NA           NA
2: North Carolina           NA          NA     0.0331327     0.021744
```

Drop:

- North Dakota from divorce rate estimation.
- North Carolina from marriage rate estimation.

```
d3L <- qread(FPath("save", "d3L.qs"))
m3L <- qread(FPath("save", "m3L.qs"))
dvdrop <- sttW[!is.na(kpss.divorce), StateName]
mrdrop <- sttW[!is.na(kpss.marriage), StateName]
d4L <- d3L[!(StateName %in% dvdrop), ]
m4L <- m3L[!(StateName %in% mrdrop), ]
qsave(d4L, "../save/d4L.qs")
qsave(m4L,"../save/m4L.qs")
```

Marriage rates and divorce rates

Event dates

```
fy <- fread(FPath("source", "FirstYearCompiledBySeiro.prn"))
d4L <- qread(FPath("save", "d4L.qs"))
m4L <- qread(FPath("save", "m4L.qs"))
```

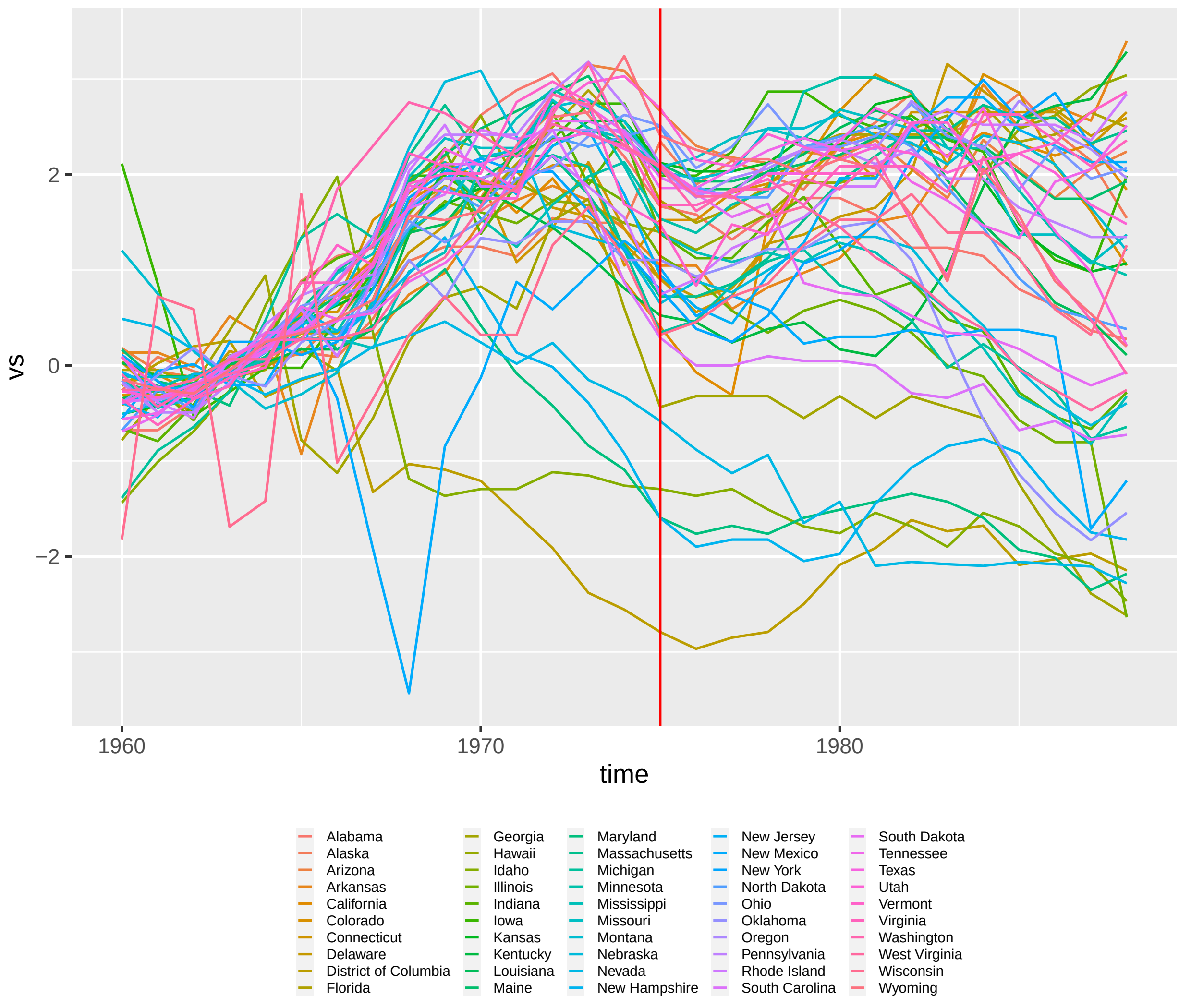

Figure 1: Marriage rates (standardized with overall std and means of 1961-1965)

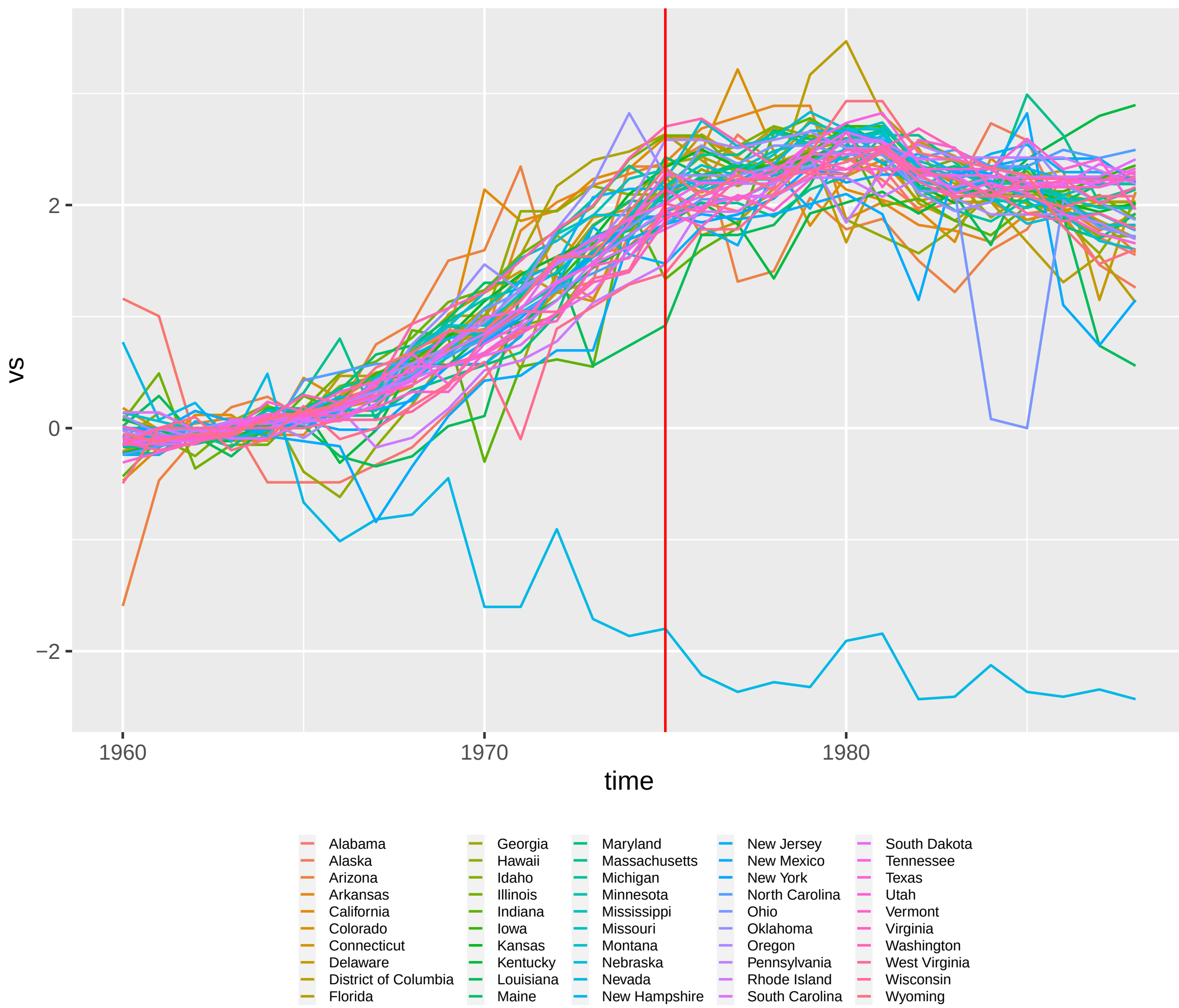

Figure 2: Divorce rates (standardized with overall std and means of 1961-1965)

```r
setnames(fy, "state", "StateName")
fy2 <- fy[, .(StateName, year, month)]
mr <- merge(m4L, fy2, by = "StateName", all = T)
dv <- merge(d4L, fy2, by = "StateName", all = T)
```

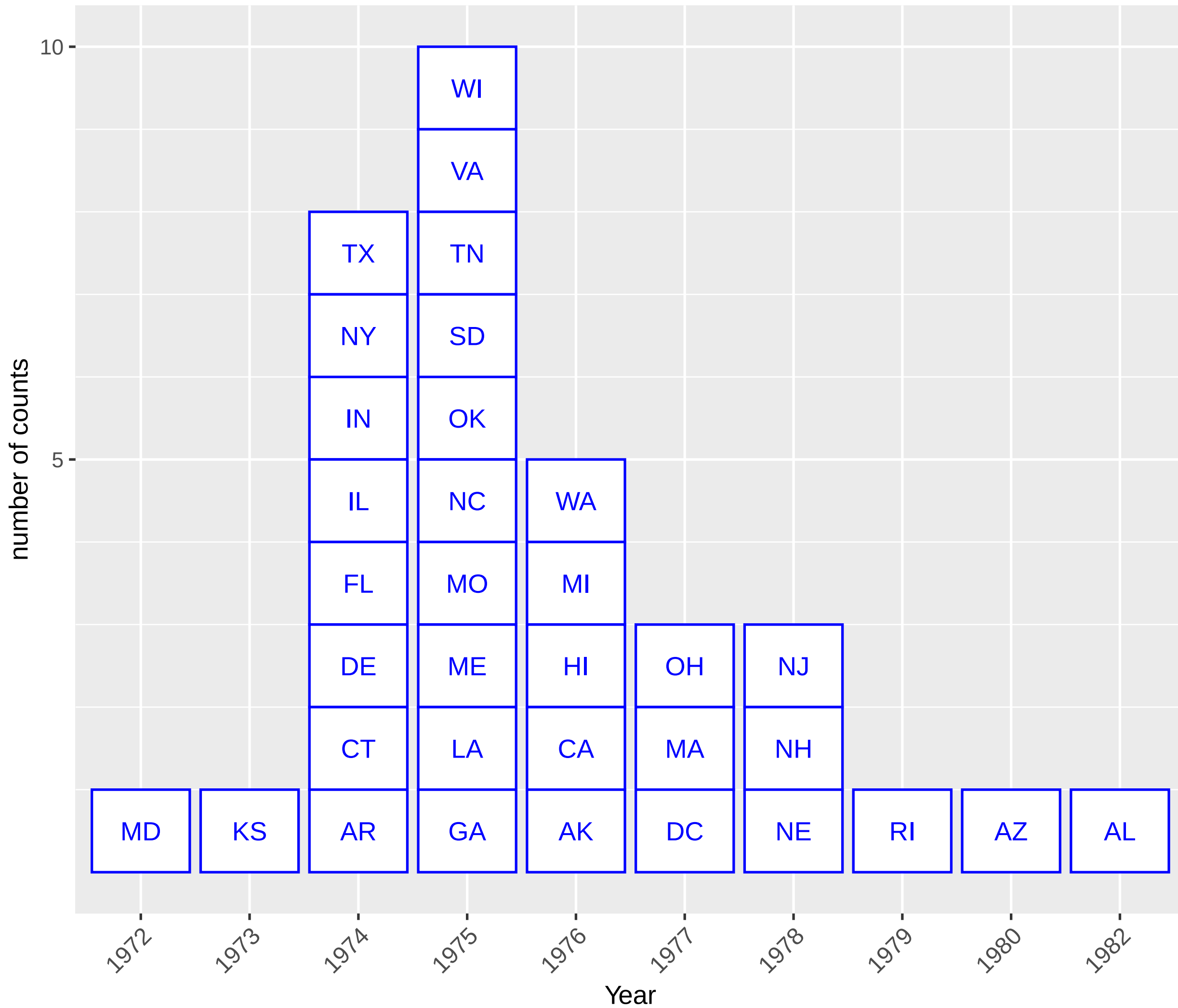

Figure 3: Event year distribution

Transform data to reflect the normalisation restriction in (c).

```r
for (ob in c("mr", "dv")) {
  obj = copy(get(ob))
  obj <- obj[!is.na(year), ]
  obj[, year := as.numeric(as.character(year))]
  obj[, time := as.numeric(as.character(time))]
  obj[, trend := time - min(time)+1, by = StateName]
  obj[, trend2 := trend^(2)]
  obj[, trend3 := trend^(3)]
  #### Normalization: At t-1, zero effect
  #### year is the year of first case in each state
```

```
  obj[, start := (year == time-1)]
  #### et: event time
  obj[, et := 1:.N, by = StateName]
  obj[, et := et-et[start], by = StateName]
  #### ptrend: pre-trend (after et=0, pre-trend is constant at ptrend[et==0])
  obj[, ptrend := trend]
  obj[, ptrend0 := trend[et==0], by = StateName]
  obj[et >= 0, ptrend := ptrend0]
  obj[, ptrend2 := trend2]
  obj[, ptrend20 := trend2[et==0], by = StateName]
  obj[et >= 0, ptrend2 := ptrend20]
  obj[, ptrend3 := trend3]
  obj[, ptrend30 := trend3[et==0], by = StateName]
  obj[et >= 0, ptrend3 := ptrend30]
  obj[, c("ptrend0", "ptrend20", "ptrend30") := NULL]
  qsave(obj, paste0(pathsave, ob, ".qs"))
  #### Normalization: mean of trend at event time < -1 is zero
  #### For this operation, keep dummy data matrix separately as etdum.
  etdum <- makeDummyFromFactor(factor(obj[, et]), nameprefix = "et")
   #### change to easier-to-handle names
  setnames(etdum, grepout("-", colnames(etdum)),
    gsub("-", "N", grepout("-", colnames(etdum))))
   #### Subtract t-L (set L=10) period to impose \bar{gamma}_{s<0} = 0
  negtime <- grepout("N", colnames(etdum))
  etdum[, (negtime) := lapply(.SD, function(x) x-etN10), .SDcols = negtime]
  etdum[, etN10 := NULL]
  #### Forcing mannually a specific order in factor levels.
  #### lm drops the first factor level as a reference.
  #### (can also be done using library(forcats), but not necessary)
   #### et: -1, -22 (or -23), -21 (or -20), ..., -2, 0, 1, ...
   #### time: 1988, 1958, 1959, ..., 1987.
   #### StateName: Hawaii, Alabama, ..., Washington, Florida
  obj[, et := factor(et, levels = c(-1, unique(et)[!(unique(et) %in% -1)]))]
  obj[, time := factor(time,
    levels = c(1988, unique(time)[!(unique(time) %in% c(1988, 1987))], 1987))]
  obj[, StateName := factor(StateName,
    levels = c("Hawaii",
      unique(StateName)[!(unique(StateName) %in% c("Hawaii", "Florida"))], "Florida"))]
  assign(ob, obj)
  assign(paste0(ob, "et"), etdum)
}
```

A technical note on how R's `lm` works using simulated data

Multiple factors

In a regression with no intercept with mutilple factor variables, there is a rule in the choice of reference levels in `lm`.

- The first factor variable in the formula uses all factor levels.
- Other factor variables in the formula drop each of the first level.

```r
set.seed(100)
#### 10 groups (a, .., t), 60 periods
dm1 <- factor(rep(letters[1:10], each = 60))
trend <- rep(1:60, 10)
dmf <- NULL
for (gg in 1:10) {
  dm1 <- letters[gg]
  trend <- 1:60
  dm2 <- factor(sample(1:4, 60, replace = T))
  dm3 <- factor(sample(1:4, 60, replace = T))
  dm4 <- factor(sample(1:4, 60, replace = T))
  dm5 <- factor(sample(1:4, 60, replace = T))
  dm6 <- factor(sample(1:4, 60, replace = T))
  dmf0 <- data.table(y=trend+as.numeric(dm2)*9-as.numeric(dm3)*3
    -as.numeric(dm4)*6+as.numeric(dm5)*2
    -as.numeric(dm3)*1.5+rnorm(60, 0, 10),
    id=dm1, trend, dm2, dm3, dm4, dm5, dm6)
  dmf <- rbind(dmf, dmf0)
}
dmf[, id := factor(id)]
dm <- lapply(dmf[, -c(1, 3)], makeDummyFromFactor,
  reference = NULL, nameprefix = "")
lapply(2:length(dm), function(i)
  setnames(dm[[i]], paste0("d", i, colnames(dm[[i]]))))
```

```r
dm <- data.table(Reduce(cbind, dm))
summary(dm)
```

```
Rank is 25, number of columns is 30,
need to drop one level from each 5 variables
in a matrix of 5 dummy variables.
```

```r
dm[, y := dmf[, y]]
lmd0 <- lm(y ~ -1 + id + dm2 + dm3 + dm4 + dm5 + dm6, data = dmf)
DFInlm <- summary(lmd0)$df
#### Taken from stats:::print.summary.lm
if (nsingular <- DFInlm[3L] - DFInlm[1L])
  cat("\nCoefficients: (",
    nsingular, " not defined because of singularities)\n")
```

Multiple factors with interactions

```r
lmd1 <- lm(y ~ -1 + id:dm2:dm3 + dm4 + dm5 + dm6, data = dmf)
DFInlm <- summary(lmd1)$df
#### Taken from stats:::print.summary.lm
if (nsingular <- DFInlm[3L] - DFInlm[1L])
  cat("\nCoefficients: (",
    nsingular, " not defined because of singularities)\n")
```

```
Coefficients: ( 4  not defined because of singularities)
```

```r
lmdc <- lmd1$coefficients
allco <- as.vector(unlist(
  unique(dmf[, .(int=as.character(interaction(id, dm2, dm3, sep = ":")))][order(int)])))
```

```r
allco <- gsub("(.):(.):", "id\\1:dm2\\2:dm3", allco)
#### Lacking 3 in dm3
dmf[id == "h" & dm2 == 2, .(id, dm2, dm3)]
```

```
   id dm2 dm3
1:  h   2   2
2:  h   2   2
3:  h   2   1
4:  h   2   4
5:  h   2   1
6:  h   2   4
7:  h   2   2
8:  h   2   1
```

Singularity is caused by lack of multiple observations for a particular combination of interactions.

Double interactions attempted by lm

- ida:dm21:dm31, idb:dm21:dm31, idc:dm21:dm31, idd:dm21:dm31, ide:dm21:dm31, idf:dm21:dm31, idg:dm21:dm31, idh:dm21:dm31, idi:dm21:dm31, idj:dm21:dm31, ida:dm22:dm31, idb:dm22:dm31, idc:dm22:dm31, idd:dm22:dm31, ide:dm22:dm31, idg:dm22:dm31, idh:dm22:dm31, idi:dm22:dm31, idj:dm22:dm31, ida:dm23:dm31, idb:dm23:dm31, idc:dm23:dm31, idd:dm23:dm31, ide:dm23:dm31, idf:dm23:dm31, idg:dm23:dm31, idh:dm23:dm31, idi:dm23:dm31, idj:dm23:dm31, ida:dm24:dm31, idb:dm24:dm31, idc:dm24:dm31, idd:dm24:dm31, ide:dm24:dm31, idf:dm24:dm31, idg:dm24:dm31, idh:dm24:dm31, idi:dm24:dm31, idj:dm24:dm31, ida:dm21:dm32, idb:dm21:dm32, idc:dm21:dm32, idd:dm21:dm32, ide:dm21:dm32, idf:dm21:dm32, idg:dm21:dm32, idh:dm21:dm32, idi:dm21:dm32, idj:dm21:dm32, ida:dm22:dm32, idb:dm22:dm32, idc:dm22:dm32, idd:dm22:dm32, ide:dm22:dm32, idf:dm22:dm32, idg:dm22:dm32, idh:dm22:dm32, idi:dm22:dm32, idj:dm22:dm32, ida:dm23:dm32, idb:dm23:dm32, idc:dm23:dm32, idd:dm23:dm32, ide:dm23:dm32, idf:dm23:dm32, idg:dm23:dm32, idh:dm23:dm32, idi:dm23:dm32, idj:dm23:dm32, ida:dm24:dm32, idb:dm24:dm32, idc:dm24:dm32, idd:dm24:dm32, ide:dm24:dm32, idf:dm24:dm32, idg:dm24:dm32, idh:dm24:dm32, idi:dm24:dm32, idj:dm24:dm32, ida:dm21:dm33, idb:dm21:dm33, idc:dm21:dm33, idd:dm21:dm33, ide:dm21:dm33, idf:dm21:dm33, idg:dm21:dm33, idh:dm21:dm33, idi:dm21:dm33, idj:dm21:dm33, ida:dm22:dm33, idb:dm22:dm33, idc:dm22:dm33, idd:dm22:dm33, ide:dm22:dm33, idf:dm22:dm33, idg:dm22:dm33, idi:dm22:dm33, idj:dm22:dm33, ida:dm23:dm33, idb:dm23:dm33, idc:dm23:dm33, idd:dm23:dm33, ide:dm23:dm33, idf:dm23:dm33, idg:dm23:dm33, idh:dm23:dm33, idi:dm23:dm33, idj:dm23:dm33, ida:dm24:dm33, idb:dm24:dm33, idc:dm24:dm33, idd:dm24:dm33, ide:dm24:dm33, idf:dm24:dm33, idg:dm24:dm33, idh:dm24:dm33, idi:dm24:dm33, idj:dm24:dm33, ida:dm21:dm34, idb:dm21:dm34, idc:dm21:dm34, idd:dm21:dm34, ide:dm21:dm34, idf:dm21:dm34, idg:dm21:dm34, idh:dm21:dm34, idi:dm21:dm34, idj:dm21:dm34, ida:dm22:dm34, idb:dm22:dm34, idc:dm22:dm34, idd:dm22:dm34, ide:dm22:dm34, idf:dm22:dm34, idg:dm22:dm34, idh:dm22:dm34, idj:dm22:dm34, ida:dm23:dm34, idb:dm23:dm34, idc:dm23:dm34, idd:dm23:dm34, ide:dm23:dm34, idf:dm23:dm34, idg:dm23:dm34, idh:dm23:dm34, idi:dm23:dm34, idj:dm23:dm34, ida:dm24:dm34, idb:dm24:dm34, idc:dm24:dm34, idd:dm24:dm34, ide:dm24:dm34, idf:dm24:dm34, idg:dm24:dm34, idh:dm24:dm34, idi:dm24:dm34, idj:dm24:dm34.

All possible double interactions

- ida:dm21:dm31, ida:dm21:dm32, ida:dm21:dm33, ida:dm21:dm34, ida:dm22:dm31, ida:dm22:dm32, ida:dm22:dm33, ida:dm22:dm34, ida:dm23:dm31, ida:dm23:dm32, ida:dm23:dm33, ida:dm23:dm34, ida:dm24:dm31, ida:dm24:dm32, ida:dm24:dm33, ida:dm24:dm34, idb:dm21:dm31, idb:dm21:dm32, idb:dm21:dm33, idb:dm21:dm34, idb:dm22:dm31, idb:dm22:dm32, idb:dm22:dm33, idb:dm22:dm34, idb:dm23:dm31, idb:dm23:dm32, idb:dm23:dm33, idb:dm23:dm34, idb:dm24:dm31, idb:dm24:dm32, idb:dm24:dm33, idb:dm24:dm34, idc:dm21:dm31, idc:dm21:dm32, idc:dm21:dm33, idc:dm21:dm34, idc:dm22:dm31, idc:dm22:dm32, idc:dm22:dm33, idc:dm22:dm34, idc:dm23:dm31, idc:dm23:dm32, idc:dm23:dm33, idc:dm23:dm34, idc:dm24:dm31, idc:dm24:dm32, idc:dm24:dm33, idc:dm24:dm34, idd:dm21:dm31, idd:dm21:dm32, idd:dm21:dm33, idd:dm21:dm34, idd:dm22:dm31, idd:dm22:dm32, idd:dm22:dm33, idd:dm22:dm34, idd:dm23:dm31, idd:dm23:dm32, idd:dm23:dm33, idd:dm23:dm34, idd:dm24:dm31, idd:dm24:dm32, idd:dm24:dm33, idd:dm24:dm34, ide:dm21:dm31, ide:dm21:dm32, ide:dm21:dm33, ide:dm21:dm34, ide:dm22:dm31, ide:dm22:dm32, ide:dm22:dm33, ide:dm22:dm34, ide:dm23:dm31, ide:dm23:dm32, ide:dm23:dm33, ide:dm23:dm34, ide:dm24:dm31, ide:dm24:dm32, ide:dm24:dm33, ide:dm24:dm34, idf:dm21:dm31, idf:dm21:dm32, idf:dm21:dm33, idf:dm21:dm34, idf:dm22:dm32, idf:dm22:dm33, idf:dm22:dm34, idf:dm23:dm31, idf:dm23:dm32, idf:dm23:dm33, idf:dm23:dm34, idf:dm24:dm31, idf:dm24:dm32, idf:dm24:dm33, idf:dm24:dm34, idg:dm21:dm31, idg:dm21:dm32, idg:dm21:dm33, idg:dm21:dm34, idg:dm22:dm31, idg:dm22:dm32, idg:dm22:dm33, idg:dm22:dm34, idg:dm23:dm31, idg:dm23:dm32, idg:dm23:dm33, idg:dm23:dm34, idg:dm24:dm31, idg:dm24:dm32, idg:dm24:dm33, idg:dm24:dm34, idh:dm21:dm31, idh:dm21:dm32, idh:dm21:dm33, idh:dm21:dm34, idh:dm22:dm31, idh:dm22:dm32, idh:dm22:dm34, idh:dm23:dm31, idh:dm23:dm32, idh:dm23:dm33, idh:dm23:dm34, idh:dm24:dm31, idh:dm24:dm32, idh:dm24:dm33, idh:dm24:dm34, idi:dm21:dm31, idi:dm21:dm32, idi:dm21:dm33, idi:dm21:dm34, idi:dm22:dm31, idi:dm22:dm32, idi:dm22:dm33, idi:dm23:dm31, idi:dm23:dm32, idi:dm23:dm33, idi:dm23:dm34, idi:dm24:dm31, idi:dm24:dm32, idi:dm24:dm33, idi:dm24:dm34, idj:dm21:dm31, idj:dm21:dm32, idj:dm21:dm33, idj:dm21:dm34, idj:dm22:dm31, idj:dm22:dm32, idj:dm22:dm33, idj:dm22:dm34, idj:dm23:dm31, idj:dm23:dm32, idj:dm23:dm33, idj:dm23:dm34, idj:dm24:dm31, idj:dm24:dm32, idj:dm24:dm33, idj:dm24:dm34.

Dropped from formula before regression (not attempted) in care of singularity

- .

Dropped ex post due to singularity (despite attempted)

- idf:dm22:dm31, idh:dm22:dm33, idi:dm22:dm34, idj:dm24:dm34 which is the same as idj:dm24:dm34.

The above is coded as `names(lmdc[is.na(lmdc)])` which is the same as `allco[allco %in% names(lmdc[is.na(lmdc)])]`.

Multiple factors with trend and interactions

```
#### Add time trend, group wise time trend
setkey(dmf, id, trend)
for (aa in letters[1:20]) {
  gt <- paste0(aa, "t")
  dmf[, (gt) := 0L]
  dmf[grepl(aa, id), (gt) := 1:.N]
}
dmf
lmd2 <- lm(y ~ -1 + id*trend*dm2 + id*trend*dm3 + id*trend*dm4
  + id*trend*dm5 + id*trend*dm6, data = dmf)
DFInlm <- summary(lmd2)$df
if (nsingular <- DFInlm[3L] - DFInlm[1L])
  cat("\nCoefficients: (",
    nsingular, " not defined because of singularities)\n")
```

In a regression with no intercept, id, trend, many factors: When id, trend, all factor variables (dm2, …, dm6; all with 5 levels) are interacted, `lm`:

- Keeps all levels of `id`.
- Drops the first level of all factors.[*3]

[*3] E.g., the first level (`dm21`) of `dm2`.

- Drops all interaction terms with dropped factor levels.[*4]
- Drops any double interaction terms using dropped factor levels.[*5]
- Drops the first level of interaction terms.[*6]
- (And keeps all other interaction terms.)

```
lmdc <- lmd2$coefficients
allco <- as.vector(unlist(
  unique(dmf[, .(int=as.character(interaction(id, dm2, sep = ":")))
  ][order(int)])))
allco <- gsub("(.):", "id\\1:trend:dm2", allco)
```

Dropped for `dm2`: Anything with `dm21` and anything with `ida:trend`.

Double interactions involving `dm2` attempted by lm

- idb:trend:dm22, idc:trend:dm22, idd:trend:dm22, ide:trend:dm22, idf:trend:dm22, idg:trend:dm22, idh:trend:dm22, idi:trend:dm22, idj:trend:dm22, idb:trend:dm23, idc:trend:dm23, idd:trend:dm23, ide:trend:dm23, idf:trend:dm23, idg:trend:dm23, idh:trend:dm23, idi:trend:dm23, idj:trend:dm23, idb:trend:dm24, idc:trend:dm24, idd:trend:dm24, ide:trend:dm24, idf:trend:dm24, idg:trend:dm24, idh:trend:dm24, idi:trend:dm24, idj:trend:dm24.

All possible double interactions involving `dm2`

- ida:trend:dm21, ida:trend:dm22, ida:trend:dm23, ida:trend:dm24, idb:trend:dm21, idb:trend:dm22, idb:trend:dm23, idb:trend:dm24, idc:trend:dm21, idc:trend:dm22, idc:trend:dm23, idc:trend:dm24, idd:trend:dm21, idd:trend:dm22, idd:trend:dm23, idd:trend:dm24, ide:trend:dm21, ide:trend:dm22, ide:trend:dm23, ide:trend:dm24, idf:trend:dm21, idf:trend:dm22, idf:trend:dm23, idf:trend:dm24, idg:trend:dm21, idg:trend:dm22, idg:trend:dm23, idg:trend:dm24, idh:trend:dm21, idh:trend:dm22, idh:trend:dm23, idh:trend:dm24, idi:trend:dm21, idi:trend:dm22, idi:trend:dm23, idi:trend:dm24, idj:trend:dm21, idj:trend:dm22, idj:trend:dm23, idj:trend:dm24.

Dropped from formula before regression (not attempted) in care of singularity

- ida:trend:dm21, ida:trend:dm22, ida:trend:dm23, ida:trend:dm24, idb:trend:dm21, idc:trend:dm21, idd:trend:dm21, ide:trend:dm21, idf:trend:dm21, idg:trend:dm21, idh:trend:dm21, idi:trend:dm21, idj:trend:dm21.

Dropped ex post due to singularity (despite attempted) (none)

- which is the same as .

ida:trend:dm22, ida:trend:dm23, ida:trend:dm24 are the dropped terms to avoid collinearity between `id`, `trend`, `dm2Y` for Y=2, …, 4.

Effects of normalisation choice

If we use `et` variable as the first regressor, `lm` uses all levels of `et`. This overparameterises the model and gives rise to multicollinearity. In such case, we need to drop one more event time manually.

Click here to see how reference period choice affects estimated results.

[*4] E.g., `id:dm21`(=`ida:dm21`, .., `idj:dm21`), `trend:dm21`, `id:dm31`, `trend:dm31`.

[*5] Anything with `id21`, `id31`, `id41`, `id51`, e.g., `idj:dm51:trend`.

[*6] E.g., `id:trend` is collinear with `id` and `trend` unless `ida:trend` is dropped. For Y=2,…5, `id:dmXY` is collinear with `id` and `dmXY` unless `ida:dmXY` is dropped. For Y=2,…5, `id:trend:dmX` is collinear with `id`, `trend` and `dmXY` unless `ida:trend:dmXY` is dropped.

Baseline $\delta_{-1} = 0$ vs. Miller recommend $\bar{\delta}_{<0} = 0$

```
summary(dv[, .(StateName, v, et, time, trend, trend2, trend3)])
```

```
      StateName          v              et            time          trend     
 Hawaii    : 30   Min.   :0.40   -1     : 34   1988    : 34   Min.   : 1.0  
 Alabama   : 30   1st Qu.:2.60   -14    : 34   1959    : 34   1st Qu.: 8.0  
 Alaska    : 30   Median :3.90   -13    : 34   1960    : 34   Median :15.5  
 Arizona   : 30   Mean   :4.05   -12    : 34   1961    : 34   Mean   :15.5  
 Arkansas  : 30   3rd Qu.:5.30   -11    : 34   1962    : 34   3rd Qu.:23.0  
 California: 30   Max.   :9.30   -10    : 34   1963    : 34   Max.   :30.0  
 (Other)   :840                  (Other):816   (Other):816                  
     trend2         trend3     
 Min.   :  1   Min.   :    1  
 1st Qu.: 64   1st Qu.:  512  
 Median :240   Median : 3736  
 Mean   :315   Mean   : 7208  
 3rd Qu.:529   3rd Qu.:12167  
 Max.   :900   Max.   :27000  
```

```
obj = copy(dv)
#### In regression with no intercept,
#### lm keeps all levels in the 1st factor variable in the formula.
#### lm drops 1st levels in the 2nd factor variable in the formula.
#### lm drops 1st and last levels in the 3rd factor variable in the formula.
 #### event time, factors
#### "v" is divorce/marriage rate (not standardised)
r10a <- lm(v ~ -1+et+StateName+time, data = obj)
r10b <- lm(v ~ -1+StateName+et+time, data = obj)
r10c <- lm(v ~ -1+StateName+time+et, data = obj)
obj[, time := factor(time,
    levels = c(1988, levels(time)[!(levels(time) %in% c(1988, 1959:1961))], 1961:1959))]
 #### event time, factors, trends
r22a <- lm(v ~ -1+et+trend+trend2+trend3+StateName+time, data = obj)
r22b <- lm(v ~ -1+trend+trend2+trend3+StateName+et+time, data = obj)
r22c <- lm(v ~ -1+trend+trend2+trend3+StateName+time+et, data = obj)
 #### Create a dummy matrix of factor variable "et"
etdumpre <- makeDummyFromFactor(factor(obj[, et]), nameprefix = "et")
setnames(etdumpre, grepout("-", colnames(etdumpre)),
  gsub("-", "N", grepout("-", colnames(etdumpre))))
 #### Subtract t-L, L=10 period to impose \bar{gamma}_{s<0} = 0
negtime <- grepout("N", colnames(etdumpre))
etdumpre[, (negtime) := lapply(.SD, function(x) x-etN10), .SDcols = negtime]
etdumpre[, etN10 := NULL]
 #### formula terms for et dummy matrix
ettermspre <- paste(colnames(etdumpre), collapse = "+")
####obj[, StateName := factor(StateName,
####  exclude=c('Hawaii', 'Florida', 'District Of Columbia'))]
obj3 <- data.table(obj, etdumpre)
 #### factors, trends, explicit event time dummies
form1 <- paste0("v ~ -1+StateName+time+", ettermspre)
form2 <- paste0("v ~ -1+trend+StateName+time+", ettermspre)
form3 <- paste0("v ~ -1+trend+trend2+trend3+StateName+time+", ettermspre)
```

```r
r31 <- lm(as.formula(form1), data = obj3)
r32 <- lm(as.formula(form2), data = obj3)
r33 <- lm(as.formula(form3), data = obj3)
```

Compare r10a, r10b, r10c, r22a, r22b, r22c, r31, r32, r33.

```r
#### explanation of forms
form0 <- c(
  #### r10
  "et+StateName+time", "StateName+et+time", "StateName+time+et",
  #### r22
  "et+trend+trend2+trend3+StateName+time",
  "trend+trend2+trend3+StateName+et+time",
  "trend+trend2+trend3+StateName+time+et",
  #### r3X
  "StateName+time+eterms",
  "trend+StateName+time+eterms",
  "trend+trend2+trend3+StateName+time+eterms")
#### explanation of term order
forder <- c(paste(rep(c("TWFE", "TWFE trend"), each = 3),
  c("et pos 1", "et pos 2", "et pos last")),
  "TWFE premean = 0", "TWFE trend premean = 0",
  "TWFE trend3 premean = 0")
#### explanation of normalization choice
 #### TWFE and TWFE trend use default normalization of factor level order
 #### r10a: all levels of et are used, r10b: first level of et is dropped, etc.
 #### r31-r33: etN10 is dropped, r32-r33: time is dropped in favor of trend
normalization <- c(rep(c("TWFE", "TWFE trend"), each = 3),
  rep("TWFE trend premean = 0", 3))
nums <- c(rep(c(10, 22), each =3), 31:33)
Ci <- NULL
for (i in 1:9) {
  if (i < 7)
    rr <- get(paste0("r", nums[i], rep(letters[1:3], 2)[i])) else
    rr <- get(c("r31", "r32", "r33")[i-6])
  clus <- data.table(rr$model)[, StateName]
  rrc <- clx(rr, cluster = clus, returnV = T)
  clxci <- data.table(cbind(Coef = rownames(rrc$ci), rrc$est, rrc$ci))
  clxci <- rbind(clxci, t(c(-1, 0, rep(NA, 5))), use.names = F)
  clxci[, FormulaOrder := forder[i]]
  clxci[, normalisation := normalization[i]]
  Ci <- rbind(Ci, clxci)
}
Ci[, period := gsub("et", "", Coef)]
Ci <- Ci[grepl("^.?\\d", period), ]
Ci[, period := gsub("N", "-", period)]
Ci[, period := as.numeric(period)]
setcolorder(Ci,
 c("Coef", "Estimate", "Std. Error", "t value", "Pr(>|t|)", "2.5 %", "97.5 %", "period"))
setnames(Ci, c("Estimate", "2.5 %", "97.5 %"), c("beta", "CI_L", "CI_U"))
numcols <- c("beta", "CI_L", "CI_U", "period")
Ci[, (numcols) := lapply(.SD, as.numeric), .SDcols = numcols]
strcols <- colnames(Ci)[!(colnames(Ci) %in% numcols)]
Ci[, (strcols) := lapply(.SD, factor), .SDcols = strcols]
```

```
Ci[, FormulaOrder := factor(FormulaOrder, levels = forder)]
```

et pos X = `et` is positioned at X. X = 1 means `et` comes first in the formula. X = 2 means `et` comes after trend and StateName. X = last means `et` comes after trend, StateName, and time.

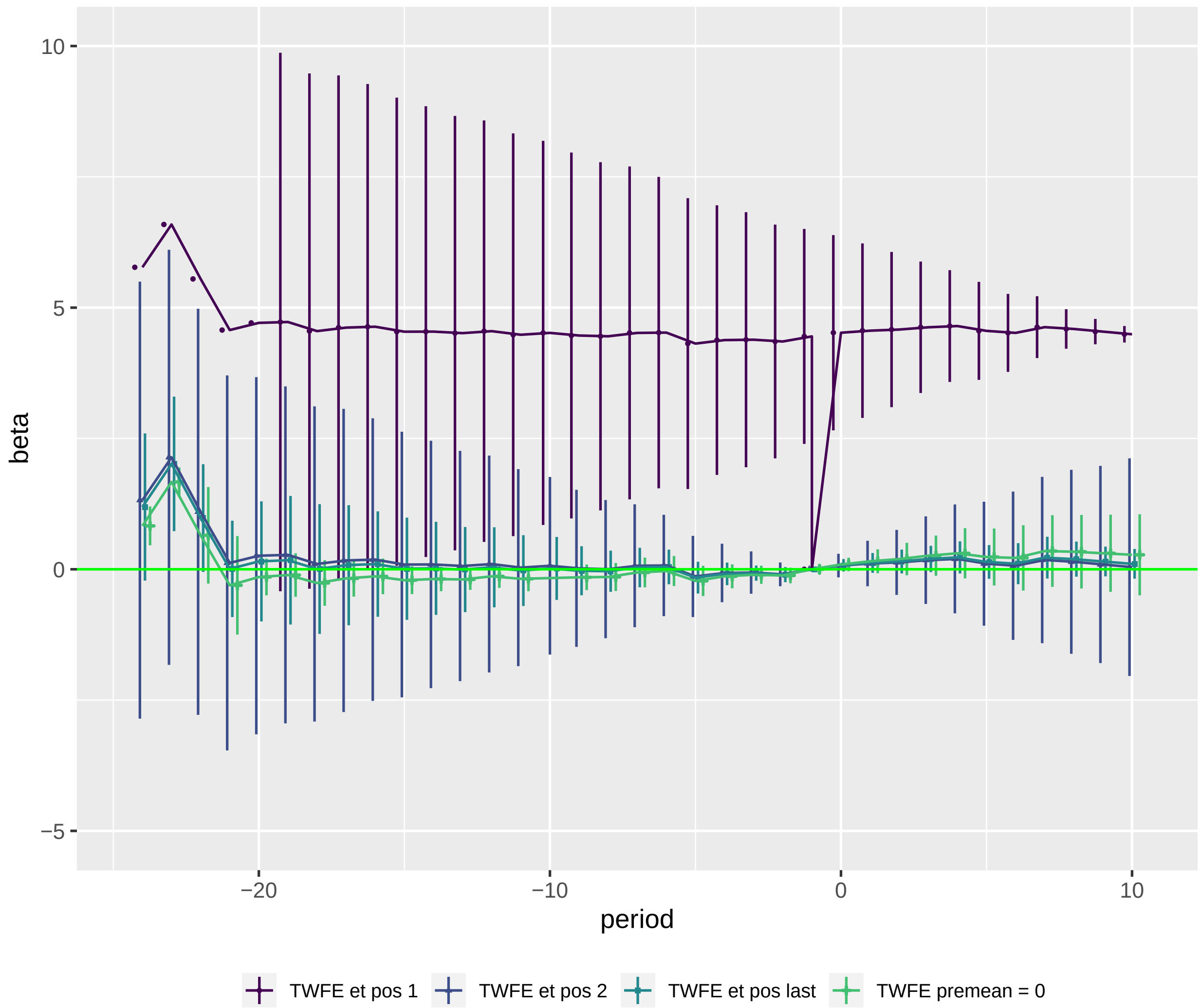

Figure 4: Impacts on divorce rates: Different normalization

Compare r10a(TWFE et pos 1), r10b(TWFE et pos 2), r10c(TWFE et pos last).

- r10a vs r10b, r10c: One sees that keeping all levels adds a value equivalent to the intercept to all estimates. This gives a problem when we force a normalisation $\gamma_{-1} = 0$ as the estimates jump around $t = -1$. Another noticeable characteristic is that standard errors decrease as event time progresses.

With trends: $\delta_{-1} = 0$ vs. $\bar{\delta}_{<0} = 0$

Compare r22a(TWFE trend et pos 1), r22b(TWFE trend et pos 2), r22c(TWFE trend et pos last), r31(TWFE premean = 0), r32(TWFE trend premean = 0), r33(TWFE trend3 premean = 0).

- r22a vs. r22b, r22c: When et assumes the role of intercept (r22a), it has to counter the rapid decline caused by the trend, thence an increasing pattern of estimated values as time passes. This must be

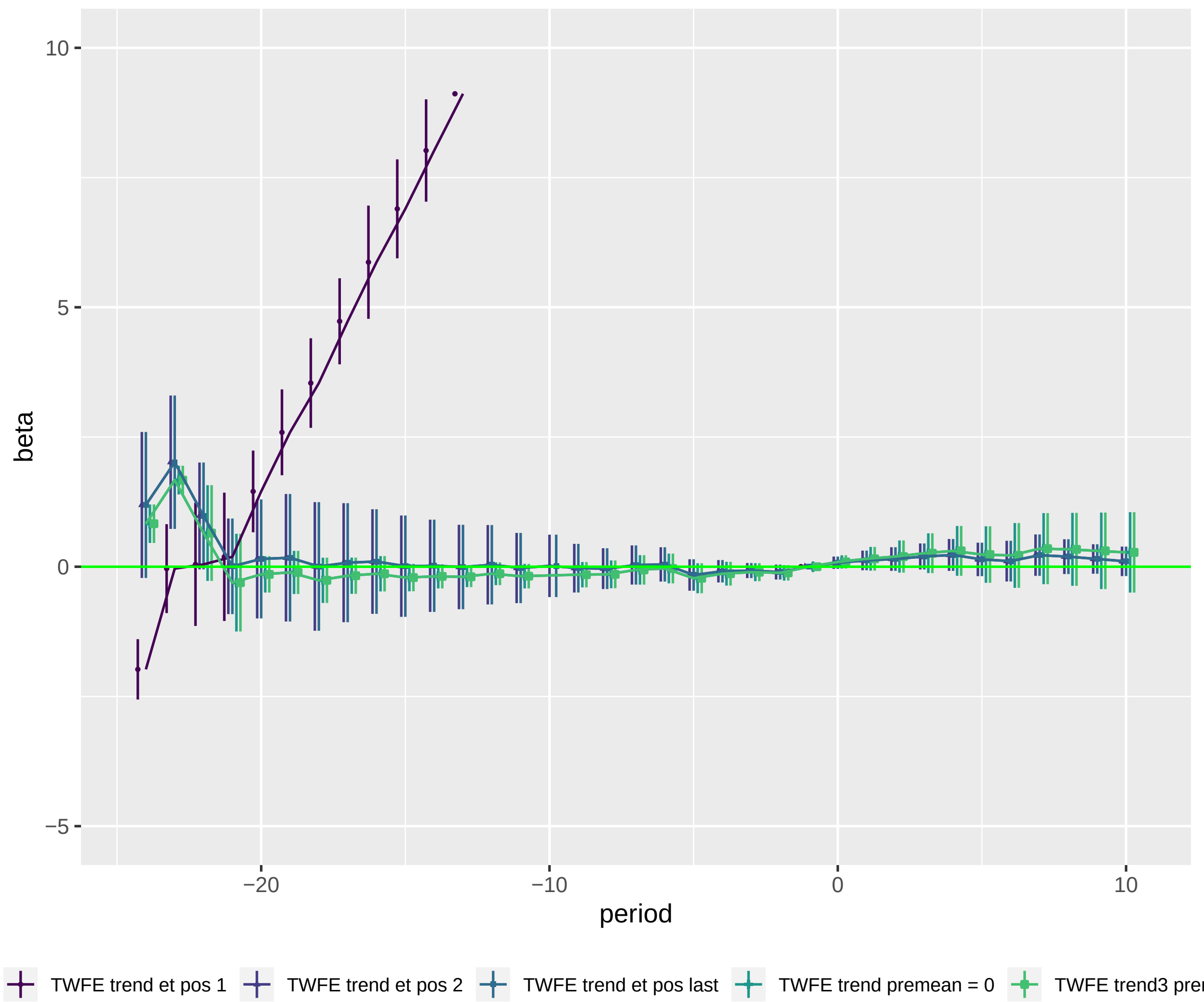

Figure 5: Impacts on divorce rates: Different normalization with trends

avoided. Both r22b and r22c drop et=-1 and et=15. All estimates of r22b and r22c are identical.

- r31: Only N10 is dropped. r32: N10 and time are dropped.

If we exclude r10a and r22a, estimates are much similar.

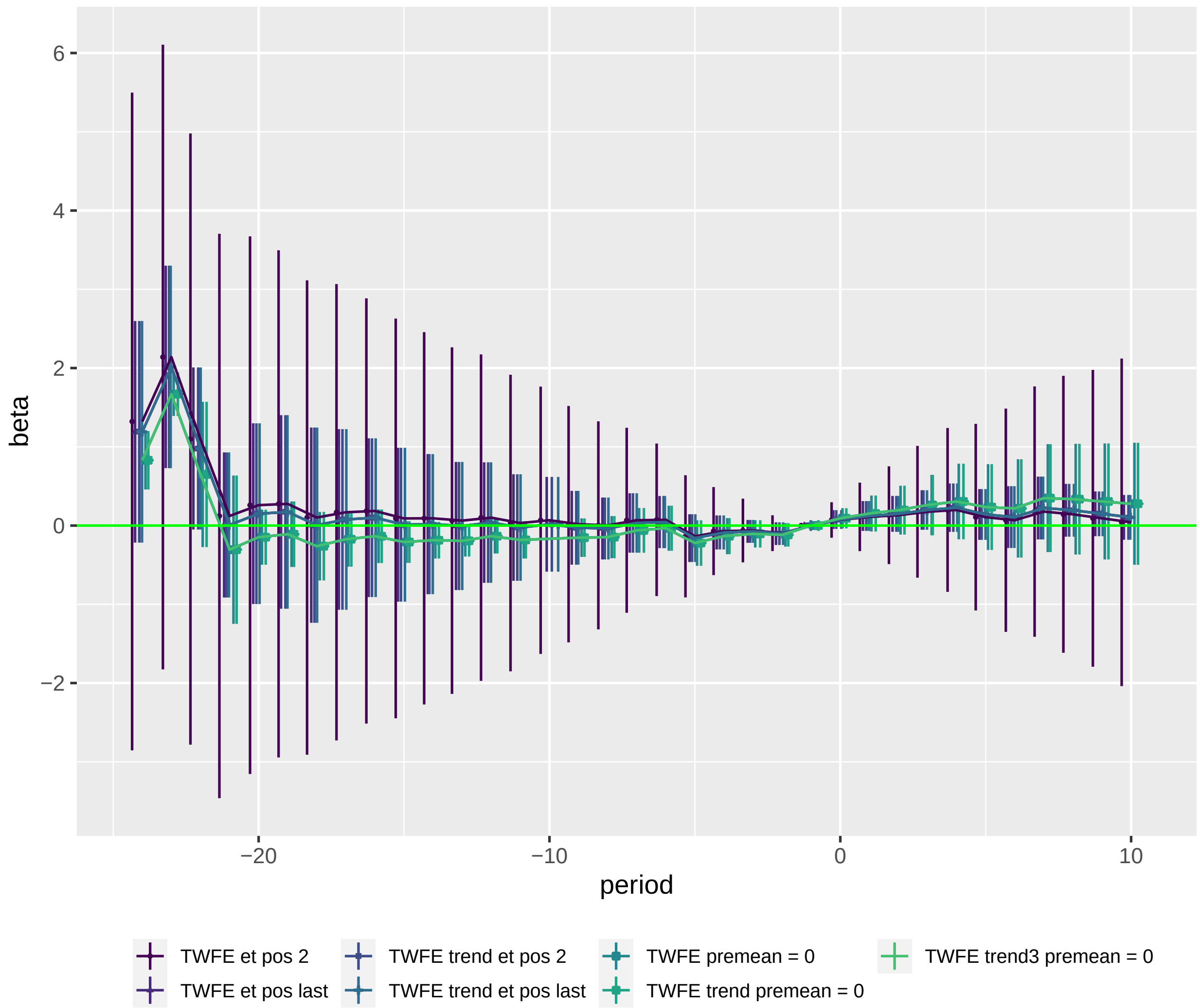

Figure 6: Impacts on divorce rates: Different normalization with event time factor defined as deviation from overall mean

- Differences in point estimates are difference in normalization choice, $\delta_{-1} = 0$ or $\bar{\delta}_{\tau<0} = 0$.

- Such seemingly an inoccuous choice has big impacts on standard errors. Premean = 0 seems to enjoy tighter CIs. This was also pointed out in Miller saying basing on one point involves larger sampling errors, but I am not sure if such reasoning is convincing.

What if we only use 1961- and etN19-et15?

```
obj = copy(dv)
r10a <- lm(v ~ -1+et+StateName+time,
  data = obj[as.numeric(as.character(time)) > 1960, ])
r10b <- lm(v ~ -1+StateName+et+time,
```

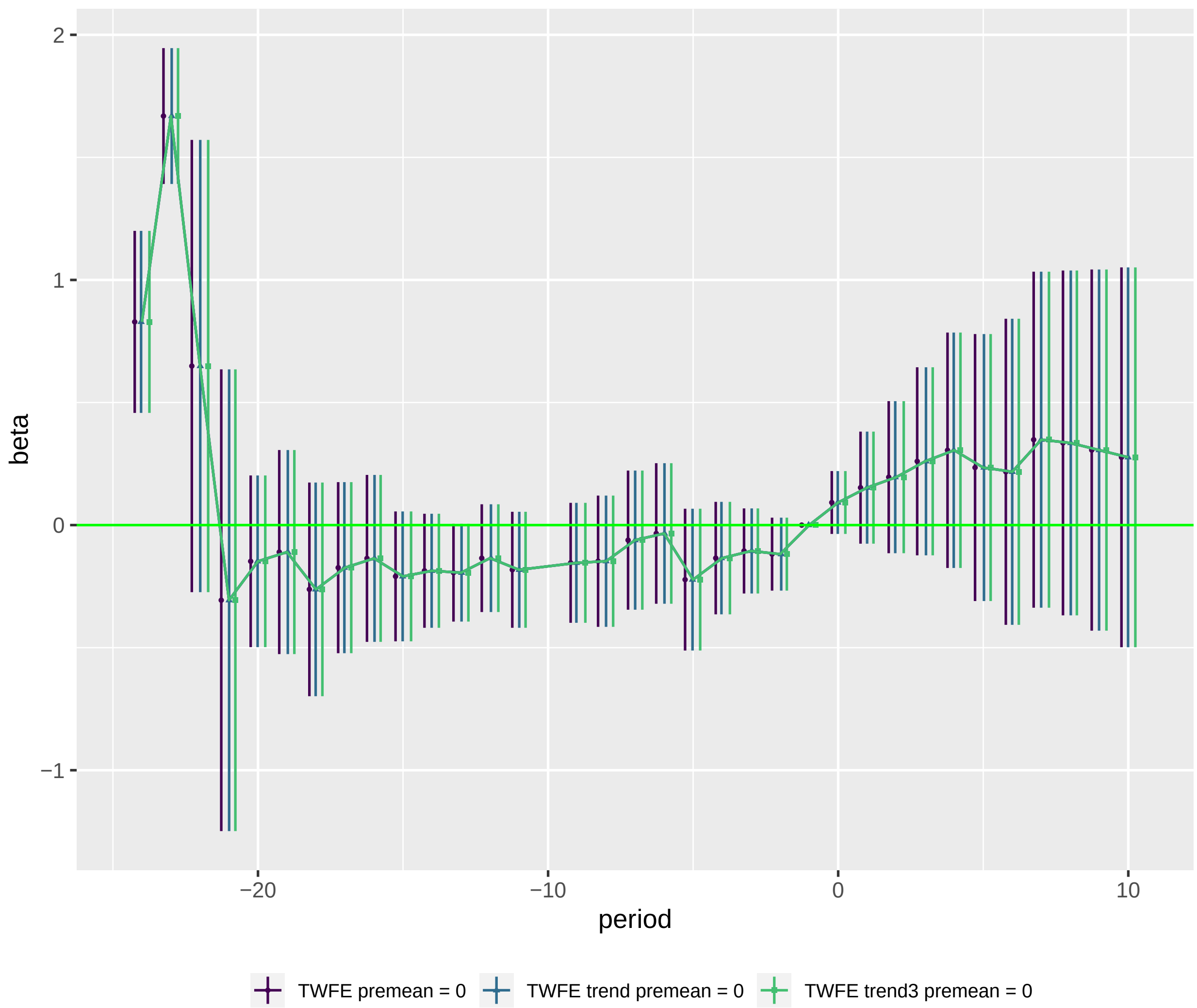

Figure 7: Impacts on divorce rates: Premean = 0 normalization

```
  data = obj[as.numeric(as.character(time)) > 1960, ])
r10c <- lm(v ~ -1+StateName+time+et,
  data = obj[as.numeric(as.character(time)) > 1960, ])
r22a <- lm(v ~ -1+et+trend+trend2+trend3+StateName+time,
  data = obj[as.numeric(as.character(time)) > 1960, ])
r22b <- lm(v ~ -1+trend+trend2+trend3+StateName+et+time,
  data = obj[as.numeric(as.character(time)) > 1960, ])
r22c <- lm(v ~ -1+trend+trend2+trend3+StateName+time+et,
  data = obj[as.numeric(as.character(time)) > 1960, ])
obj[, time := factor(time,
    levels = c(1988, levels(time)[!(levels(time) %in% c(1988, 1959:1961))], 1961:1959))]
etdumpre <- makeDummyFromFactor(factor(obj[, et]), nameprefix = "et", reference = NULL)
setnames(etdumpre, grepout("-", colnames(etdumpre)),
  gsub("-", "N", grepout("-", colnames(etdumpre))))
negtime <- grepout("N", colnames(etdumpre))
etdumpre[, (negtime) := lapply(.SD, function(x) x-etN10), .SDcols = negtime]
etdumpre[, etN10 := NULL]
ettermspre <- paste(colnames(etdumpre), collapse = "+")
obj3 <- data.table(obj, etdumpre)
ettermspre2 <- gsub("etN2..*etN20\\+", "", ettermspre)
form1 <- paste0("v ~ -1+StateName+time+", ettermspre2)
form2 <- paste0("v ~ -1+trend+StateName+time+", ettermspre2)
form3 <- paste0("v ~ -1+trend+trend2+trend3+StateName+time+", ettermspre2)
r31 <- lm(as.formula(form1), data = obj3[as.numeric(as.character(time)) > 1960, ])
r32 <- lm(as.formula(form2), data = obj3[as.numeric(as.character(time)) > 1960, ])
r33 <- lm(as.formula(form3), data = obj3[as.numeric(as.character(time)) > 1960, ])
Ci <- NULL
normalization <- c(rep(c("TWFE", "TWFE trend"), each = 3),
  rep("TWFE trend premean = 0", 3))
nums <- c(rep(c(10, 22), each =3), 31:33)
Ci <- NULL
for (i in 1:9) {
  if (i < 7)
    rr <- get(paste0("r", nums[i], rep(letters[1:3], 2)[i])) else
    rr <- get(c("r31", "r32", "r33")[i-6])
  clus <- data.table(rr$model)[, StateName]
  rrc <- clx(rr, cluster = clus, returnV = T)
  clxci <- data.table(cbind(Coef = rownames(rrc$ci), rrc$est, rrc$ci))
  if (i < 7) clxci <- rbind(clxci, t(c(-1, 0, rep(NA, 5))), use.names = F)
  clxci[, FormulaOrder := forder[i]]
  clxci[, normalisation := normalization[i]]
  Ci <- rbind(Ci, clxci)
}
Ci[, period := gsub("et", "", Coef)]
Ci <- Ci[grepl("^.?\\d|tre", period), ]
Ci[, period := gsub("N", "-", period)]
Ci[, period := as.numeric(period)]
setcolorder(Ci,
 c("Coef", "Estimate", "Std. Error", "t value", "Pr(>|t|)", "2.5 %", "97.5 %", "period"))
setnames(Ci, c("Estimate", "2.5 %", "97.5 %"), c("beta", "CI_L", "CI_U"))
numcols <- c("beta", "CI_L", "CI_U", "period")
Ci[, (numcols) := lapply(.SD, as.numeric), .SDcols = numcols]
strcols <- colnames(Ci)[!(colnames(Ci) %in% numcols)]
```

```r
Ci[, (strcols) := lapply(.SD, factor), .SDcols = strcols]
Ci[, FormulaOrder := factor(FormulaOrder, levels = forder)]
Ci[grepl("pre", FormulaOrder) & abs(period) < 2,
  .(Coef, beta, period, FormulaOrder)][order(Coef, FormulaOrder)]
```

```
   Coef     beta period            FormulaOrder
1:  et0 0.661398      0        TWFE premean = 0
2:  et0 0.661398      0  TWFE trend premean = 0
3:  et0 0.661398      0 TWFE trend3 premean = 0
4:  et1 0.765569      1        TWFE premean = 0
5:  et1 0.765569      1  TWFE trend premean = 0
6:  et1 0.765569      1 TWFE trend3 premean = 0
7: etN1 0.525515     -1        TWFE premean = 0
8: etN1 0.525515     -1  TWFE trend premean = 0
9: etN1 0.525515     -1 TWFE trend3 premean = 0
```

```r
Ci[grepl("pre", FormulaOrder) & grepl("trend", Coef),
  c("Coef", "beta", "Pr(>|t|)", "FormulaOrder")][order(Coef, FormulaOrder)]
```

```
     Coef        beta           Pr(>|t|)            FormulaOrder
1:  trend  0.02886720   0.12558331462347  TWFE trend premean = 0
2:  trend  3.96516928 0.0798549729677805 TWFE trend3 premean = 0
3: trend2 -0.22953462 0.0932236296226804 TWFE trend3 premean = 0
4: trend3  0.00364198  0.105279801102284 TWFE trend3 premean = 0
```

- Estimates are exactly the same for all models.

- Trend and time FEs have a linear relationship. This makes estimates on other variables exactly the same.

- When time FEs and trend are put together, we can define a new set of time FEs $\tilde{\tau}_1 = \tau_1 - trend$, $\tilde{\tau}_2 = \tau_2 - 2 * trend$, ... where τ_t is time t FE in the regression without a trending term. So when time FEs and trend are used as covariates, and one of τ_t must be dropped to avoid collinearity with $trend$ and τ_t.

In essence, for the divorce and marriage data sets, the only variations in estimation specification that are worths examining are:

- TWFE with $\delta_{-1} = 0$, and,

- TWFE with $\bar{\delta}_{\tau<0} = 0$.

- In both, one of `et` factor levels is dropped.
- It is essential to note the increasing trend in divorce rates when we try to understand the differences in model estimates.

- Restriction $\bar{\delta}_{\tau<0} = 0$ pulls down pre-period estimates. When combined with the increasing trend, it gives rise to elevated post-period estimates.

- Using only time FEs (TWFE) stabilizes point estimates around zero, but at the cost of increasing SEs or a less precise fit.

- With smaller SEs, one would choose the model with $\bar{\delta}_{\tau<0} = 0$ restriction.

- The last point may be more emphasized once we see that $\delta_{-1} = 0$ and $\bar{\delta}_{<0} = 0$ are a transformation

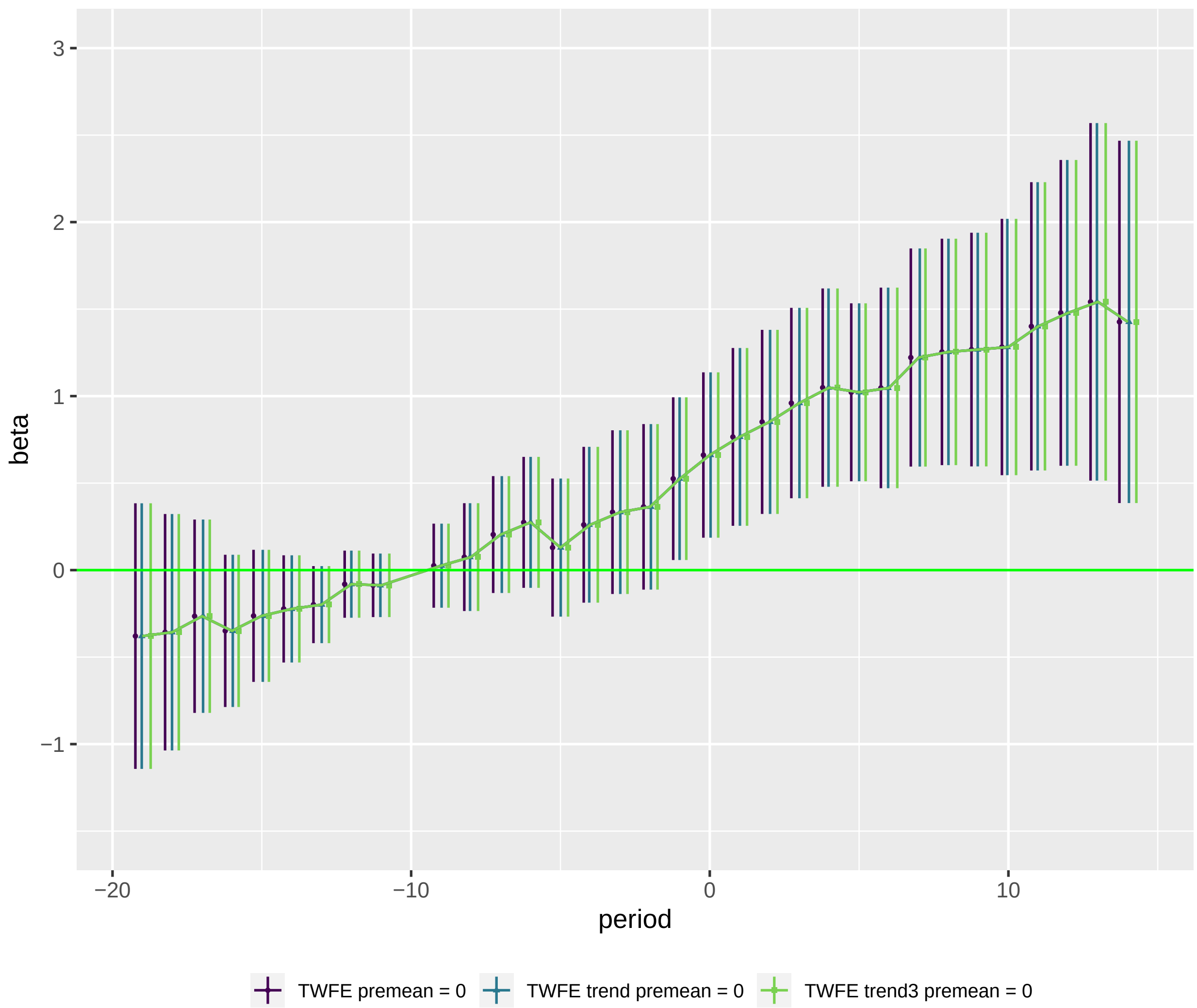

Figure 8: Impacts on divorce rates: Premean = 0 normalization, 1961-

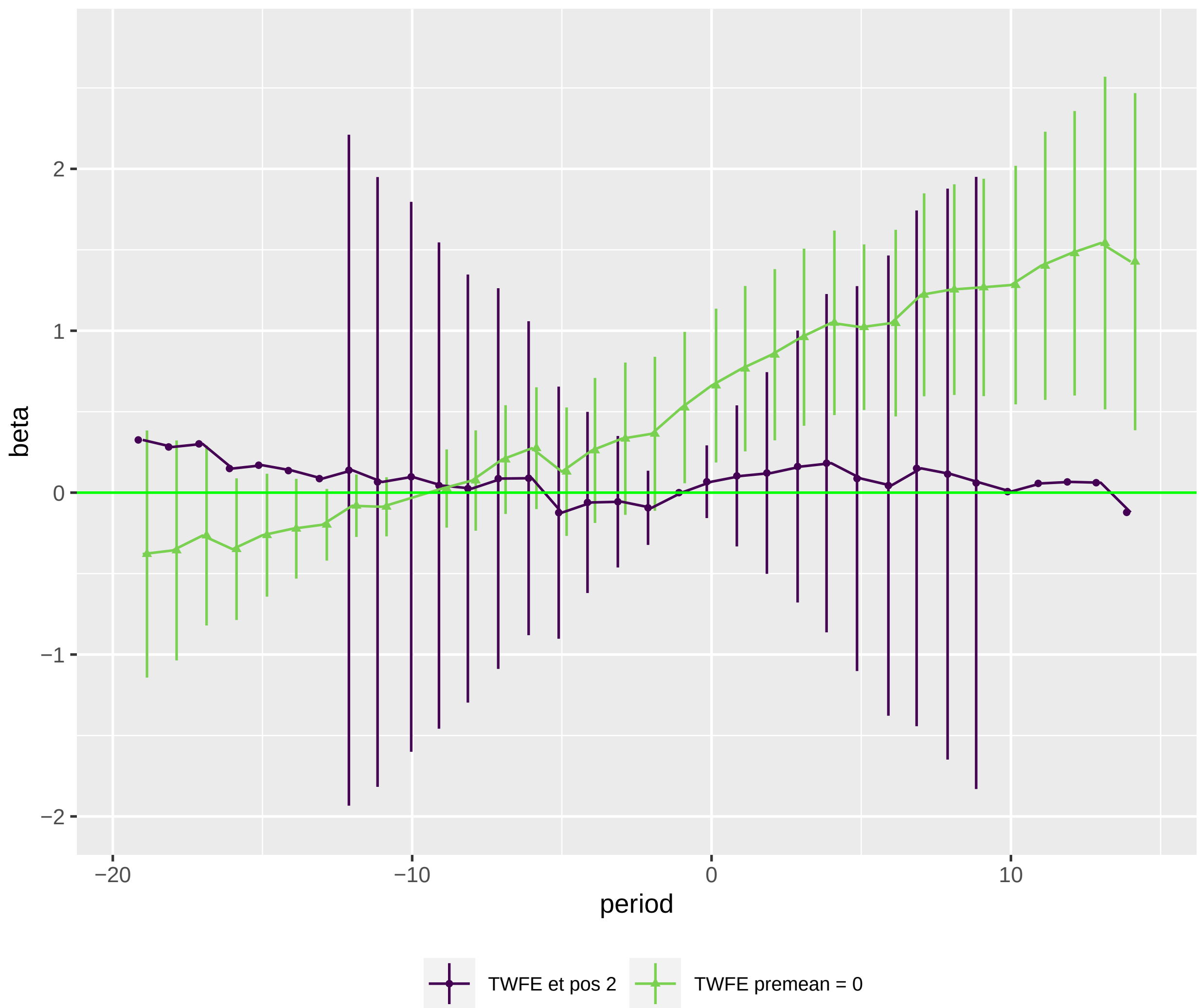

Figure 9: Impacts on divorce rates: t=-1 and premean normalization, 1961-1987

of one another: If we tilt $\delta_{-1} = 0$ (purple line) counter clockwise at around the point where $\bar{\delta}_{<0} = 0$ (green) crosses zero, we will have the same plots. This implies smaller SEs can be a free lunch.

- There seems to be a pattern in SEs: They are smallest when the sign of estimates change (when the line cross zero).
  - $\delta_{-1} = 0$ has small p values right after the impact but not later.
  - Later impacts (mechanically?) have larger p values.

State specific trends: $\delta_{-1} = 0$ vs. $\bar{\delta}_{<0} = 0$

```
Ci <- NULL
for (ob in c("mr", "dv")) {
  obj = copy(get(ob))
  obj[, et := factor(et)]
  #### r22: state individual trends, time, et with \delta_{-1}=0.
  r22 <- lm(v ~ -1+trend*StateName+time+et,
    data = obj[as.numeric(as.character(time)) > 1960, ])
  obj[, time := factor(time,
      levels = c(1988, levels(time)[!(levels(time) %in% c(1988, 1959:1961))], 1961:1959))]
  etdumpre <- makeDummyFromFactor(factor(obj[, et]), nameprefix = "et", reference = NULL)
  setnames(etdumpre, grepout("-", colnames(etdumpre)),
    gsub("-", "N", grepout("-", colnames(etdumpre))))
  #### negtime: negative et periods. This is used to impose
  #### Preperiod = 0 restriction on the data matrix.
  negtime <- grepout("N", colnames(etdumpre))
  etdumpre[, (negtime) := lapply(.SD, function(x) x-etN10), .SDcols = negtime]
  etdumpre[, etN10 := NULL]
  obj3 <- data.table(obj, etdumpre)
  #### ettermspre2: et terms etN19+etN18+...+et10
  #### (with corresponding matrix etdumpre has Preperiod = 0 imposed)
  ettermspre <- paste(colnames(etdumpre), collapse = "+")
  ettermspre2 <- gsub("etN2..*etN20\\+", "", ettermspre)
  form2 <- paste0("v ~ -1+trend*StateName+time+", ettermspre2)
  #### r32: state individual trends, time, and et with preperiod = 0
  r32 <- lm(as.formula(form2), data = obj3[as.numeric(as.character(time)) > 1960, ])
  normalization <- c(rep(c("TWFE", "TWFE trend"), each = 3),
    rep("TWFE trend premean = 0", 3))
  nums <- c(rep(c(10, 22), each =3), 31:33)
  #### Below loop works on r10a, ..., r33, but we extract only r22 and r32
  #### which are estimated in this chunk.
  for (i in 1:9) {
    if (i < 7)
      rr <- get(paste0("r", nums[i], rep(letters[1:3], 2)[i])) else
      rr <- get(c("r31", "r32", "r33")[i-6])
    clus <- data.table(rr$model)[, StateName]
    rrc <- clx(rr, cluster = clus, returnV = T)
    clxci <- data.table(cbind(Coef = rownames(rrc$ci), rrc$est, rrc$ci))
    if (i < 7) clxci <- rbind(clxci, t(c(-1, 0, rep(NA, 5))), use.names = F)
    clxci[, FormulaOrder := forder[i]]
    clxci[, normalisation := normalization[i]]
    clxci[, spec := paste0("r",
      c("10a", "10b", "10c", "22a", "22", "23c", "31", "32", "33")[i])]
    clxci[, obj := ob]
```

```r
    clxci[, period := gsub("et", "", Coef)]
    clxci[, period := gsub("N", "-", period)]
    clxci[, period := as.numeric(period)]
    setnames(clxci, c("Estimate", "2.5 %", "97.5 %"), c("beta", "CI_L", "CI_U"))
    numcols <- c("beta", "CI_L", "CI_U", "period")
    clxci[, (numcols) := lapply(.SD, as.numeric), .SDcols = numcols]
    strcols <- colnames(clxci)[!(colnames(clxci) %in% numcols)]
    clxci[, (strcols) := lapply(.SD, factor), .SDcols = strcols]
    clxci[, FormulaOrder := factor(FormulaOrder, levels = forder)]
    clxci[grepl("22$|32", spec), FormulaOrder := gsub("trend", "itrend", FormulaOrder)]
    Ci <- rbind(Ci, clxci)
  }
}
setcolorder(Ci,
 c("Coef", "beta", "Std. Error", "t value", "Pr(>|t|)", "CI_L", "CI_U", "period"))
setnames(Ci, "Pr(>|t|)", "pval")
Ci[grepl("itrend pre", FormulaOrder) & grepl("trend:", Coef) & grepl("dv", obj),
  .(Coef=gsub(".StateName", "\\*", Coef), beta=round(beta, 4),
  "p(%)"=round(as.numeric(as.character(pval))*100, 2), FormulaOrder)][
    order(Coef, FormulaOrder)]
```

```
                           Coef    beta p(%)            FormulaOrder
 1:              trend*Alabama -0.7632 0.19 TWFE itrend premean = 0
 2:               trend*Alaska  0.0711 0.00 TWFE itrend premean = 0
 3:              trend*Arizona -0.5654 0.06 TWFE itrend premean = 0
 4:             trend*Arkansas  0.3164 0.01 TWFE itrend premean = 0
 5:           trend*California -0.0284 0.00 TWFE itrend premean = 0
 6:          trend*Connecticut  0.2709 0.09 TWFE itrend premean = 0
 7:             trend*Delaware  0.2978 0.03 TWFE itrend premean = 0
 8: trend*District of Columbia -0.1099 0.66 TWFE itrend premean = 0
 9:              trend*Florida  0.2599 0.15 TWFE itrend premean = 0
10:              trend*Georgia  0.1608 0.01 TWFE itrend premean = 0
11:             trend*Illinois  0.2346 0.40 TWFE itrend premean = 0
12:              trend*Indiana  0.2967 0.03 TWFE itrend premean = 0
13:               trend*Kansas  0.4003 0.11 TWFE itrend premean = 0
14:            trend*Louisiana  0.1111 0.64 TWFE itrend premean = 0
15:                trend*Maine  0.1409 0.06 TWFE itrend premean = 0
16:             trend*Maryland  0.4925 0.26 TWFE itrend premean = 0
17:        trend*Massachusetts -0.1637 0.01 TWFE itrend premean = 0
18:             trend*Michigan -0.0236 0.00 TWFE itrend premean = 0
19:             trend*Missouri  0.1178 0.39 TWFE itrend premean = 0
20:             trend*Nebraska -0.2746 0.07 TWFE itrend premean = 0
21:        trend*New Hampshire -0.2520 0.18 TWFE itrend premean = 0
22:           trend*New Jersey -0.2472 0.22 TWFE itrend premean = 0
23:             trend*New York  0.3016 0.02 TWFE itrend premean = 0
24:       trend*North Carolina  0.1644 0.01 TWFE itrend premean = 0
25:                 trend*Ohio -0.1622 0.01 TWFE itrend premean = 0
26:             trend*Oklahoma  0.1344 0.10 TWFE itrend premean = 0
27:         trend*Rhode Island -0.3861 0.15 TWFE itrend premean = 0
28:         trend*South Dakota  0.1274 0.18 TWFE itrend premean = 0
29:            trend*Tennessee  0.1818 0.00 TWFE itrend premean = 0
30:                trend*Texas  0.2551 0.18 TWFE itrend premean = 0
31:             trend*Virginia  0.1330 0.11 TWFE itrend premean = 0
32:           trend*Washington  0.0092 0.00 TWFE itrend premean = 0
```

```
33:             trend*Wisconsin  0.1201 0.32 TWFE itrend premean = 0
                            Coef    beta p(%)            FormulaOrder
```

```r
Ci[grepl("itrend pre", FormulaOrder) & grepl("trend:", Coef) & grepl("mr", obj),
  .(Coef=gsub(".StateName", "\\*", Coef), beta=round(beta, 4),
  "p(%)"=round(as.numeric(as.character(pval))*100, 2), FormulaOrder)][
    order(Coef, FormulaOrder)]
```

```
                            Coef    beta  p(%)            FormulaOrder
 1:               trend*Alabama  0.1059 50.98 TWFE itrend premean = 0
 2:                trend*Alaska -0.0078  0.00 TWFE itrend premean = 0
 3:               trend*Arizona  0.1312 22.36 TWFE itrend premean = 0
 4:              trend*Arkansas -0.1964  0.02 TWFE itrend premean = 0
 5:            trend*California -0.1488  0.00 TWFE itrend premean = 0
 6:           trend*Connecticut -0.2723  0.00 TWFE itrend premean = 0
 7:              trend*Delaware -0.2220  0.00 TWFE itrend premean = 0
 8: trend*District of Columbia -0.3075  0.00 TWFE itrend premean = 0
 9:               trend*Florida -0.2028  0.01 TWFE itrend premean = 0
10:               trend*Georgia -0.3322  0.00 TWFE itrend premean = 0
11:              trend*Illinois -0.3724  0.00 TWFE itrend premean = 0
12:               trend*Indiana -0.3452  0.00 TWFE itrend premean = 0
13:                trend*Kansas -0.3081  0.01 TWFE itrend premean = 0
14:             trend*Louisiana -0.2275  0.00 TWFE itrend premean = 0
15:                 trend*Maine -0.1939  0.00 TWFE itrend premean = 0
16:              trend*Maryland -0.5693  0.00 TWFE itrend premean = 0
17:         trend*Massachusetts -0.0986  0.02 TWFE itrend premean = 0
18:              trend*Michigan -0.2428  0.00 TWFE itrend premean = 0
19:              trend*Missouri -0.2127  0.00 TWFE itrend premean = 0
20:              trend*Nebraska -0.1273  1.50 TWFE itrend premean = 0
21:         trend*New Hampshire -0.2231  0.00 TWFE itrend premean = 0
22:            trend*New Jersey -0.0556 28.76 TWFE itrend premean = 0
23:              trend*New York -0.2669  0.00 TWFE itrend premean = 0
24:                  trend*Ohio -0.0779  0.28 TWFE itrend premean = 0
25:              trend*Oklahoma -0.3416  0.00 TWFE itrend premean = 0
26:          trend*Rhode Island  0.0057 94.22 TWFE itrend premean = 0
27:          trend*South Dakota -0.3196  0.00 TWFE itrend premean = 0
28:             trend*Tennessee -0.1466  0.00 TWFE itrend premean = 0
29:                 trend*Texas -0.2583  0.00 TWFE itrend premean = 0
30:              trend*Virginia -0.2123  0.00 TWFE itrend premean = 0
31:            trend*Washington -0.2038  0.00 TWFE itrend premean = 0
32:             trend*Wisconsin -0.2300  0.00 TWFE itrend premean = 0
                            Coef    beta  p(%)            FormulaOrder
```

```r
setnames(Ci, "pval", "Pr(>|t|)")
```

Estimates on state specific trends are large in magnitude and have very small p values. These large-in-magnitude and heterogenous state specific trends affect the estimates on `et` in an unpredicted way. Because their magnitude is large, in the current case, `et` estimates diverge (to negative infinity?).

- Estimates under premean $= 0$ with state specific trends are not sensible.
- Why are only models with $\bar{\delta}_{\tau<0} = 0$ affected?

To dig into what is going on, we will:

- Restrict et to be between -15 and 10 (or 11) because there are fewer observations outside this window

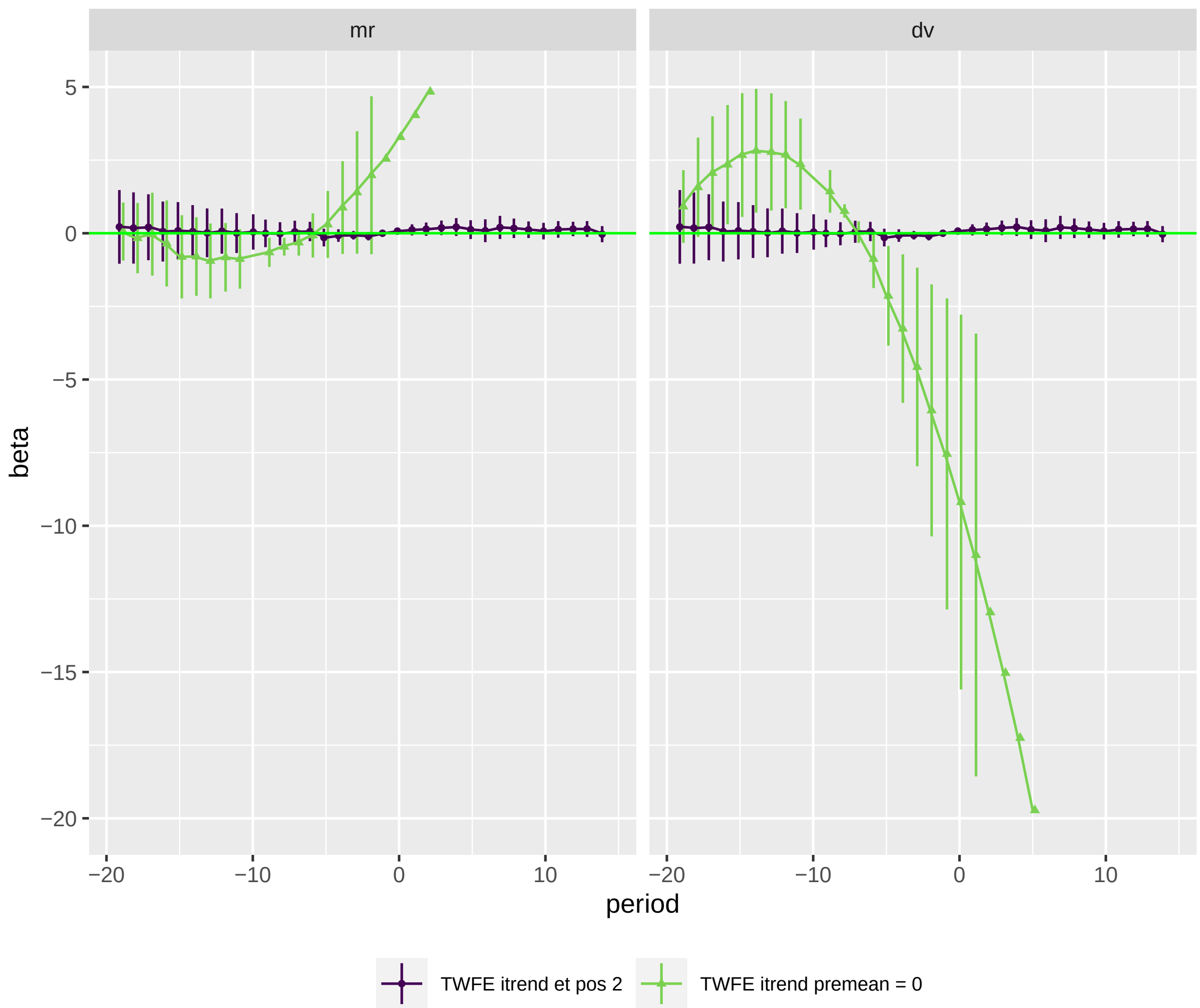

Figure 10: Impacts on divorce and marriage rates: State specific trends

(see the below table).

- Keep only a few states and see.

```
options(width = 120)
Ci <- NULL
for (ob in c("mr", "dv")) {
  obj1 = copy(get(ob))
  #### Restrict: -15 <= et <= 10
  obj1[, time := factor(time,
      levels = c(1988, levels(time)[!(levels(time) %in% c(1988, 1959:1961))],
        1961:1959))]
  obj1[, et := as.numeric(as.character(et))]
  #### et < -15 | et > 10 ==> -1, so it does not directly
  #### affect delta (et estimates) once we impose \delta_{-1}=0.
  obj1[et < -15 | et > 10, et := -1L]
  obj1[, et := factor(et)]
  obj1[, et := factor(et, levels = c(-1, levels(et)[!(levels(et) %in% c(-1))]))]
  obj1 <- obj1[grepl("^[A-Z]", StateName) & as.numeric(as.character(time)) > 1960, ]
  #### r10bb: TWFE with et window restriction
  r10bb <- lm(v ~ -1+StateName+time+et,
    data = obj1[as.numeric(as.character(time)) > 1960, ])
  #### r22: TWFE with indiv trends and et window restriction, small samples
  r22s <- lm(v ~ -1+trend*StateName+time+et, data = obj1)
  #### Explicitly drop etN16, etN17, ..., et11, et12, and allow etN1 to be kept
  etdumpre <- makeDummyFromFactor(factor(obj1[, et]),
    nameprefix = "et", reference = NULL)
  setnames(etdumpre, grepout("-", colnames(etdumpre)),
    gsub("-", "N", grepout("-", colnames(etdumpre))))
  #### Drop et < -15 | et > 10 terms
  etdumpre[, grepout("N2.|N1[6-9]|et1[1-9]", colnames(etdumpre)) := NULL]
  ettermspre <- paste(colnames(etdumpre), collapse = "+")
  #### Explicitly drop et < -15 | et > 10 terms and drop etN1
  etdumpre[, etN1 := NULL]
  ettermspre <- paste(colnames(etdumpre), collapse = "+")
  obj3 <- data.table(obj1, etdumpre)
  form2 <- paste0("v ~ -1+trend*StateName+time+", ettermspre)
  #### r22ee: TWFE with indiv trends, et window restriction, dropping N1
  #### TWFE itrend no N1 N15P10
  r22ees <- lm(as.formula(form2), data = obj3)
  #### Preperiod = 0 restriction on the data matrix.
  etdumpre <- makeDummyFromFactor(factor(obj1[, et]),
    nameprefix = "et", reference = NULL)
  setnames(etdumpre, grepout("-", colnames(etdumpre)),
    gsub("-", "N", grepout("-", colnames(etdumpre))))
  etdumpre[, grepout("N2.|N1[6-9]|et1[1-9]", colnames(etdumpre)) := NULL]
  negtime <- grepout("N", colnames(etdumpre))
  etdumpre[, (negtime) := lapply(.SD, function(x) x-etN10), .SDcols = negtime]
  etdumpre[, etN10 := NULL]
  #### Drop etN2X - etN20 from formula.
  #### This is already dropped so no change is made here.
  ettermspre <- paste(colnames(etdumpre), collapse = "+")
  ettermspre2 <- gsub("etN2..*etN20\\+", "", ettermspre)
  obj3 <- data.table(obj1, etdumpre)
```

```
  form2 <- paste0("v ~ -1+trend*StateName+time+", ettermspre2)
  #### r32s: state individual trends, time, et with preperiod = 0
  r32s <- lm(as.formula(form2), data = obj3)
  #### Add "TWFE itrend N15P10" to forder, paste N15P10 to r22
  regob <- paste0("r", c("10a", "10bb", "10c",
      "22a", "22", "22c", "31", "32s", "33", #### 32s is substituted to 32
       "22ees")) #### this is added
  forder2 <- c(forder, "TWFE itrend N15P10", "TWFE itrend no N1 N15P10")
  forder2 <- gsub("i?trend et pos 2", "itrend et pos 2 N15P10", forder2)
  forder2 <- gsub("E et pos 2", "E et pos 2 N15P10", forder2)
  forder2 <- gsub(" trend premean = 0", " itrend premean = 0 N15P10", forder2)
  Ci0 <- NULL
  for (i in 1:length(regob)) {
    rr <- get(regob[i])
    clus <- data.table(rr$model)[, StateName]
    rrc <- clx(rr, cluster = clus, returnV = T)
    clxci <- data.table(cbind(Coef = rownames(rrc$ci), rrc$est, rrc$ci))
    if (i < 7) clxci <- rbind(clxci, t(c(-1, 0, rep(NA, 5))), use.names = F)
    clxci[, FormulaOrder := forder2[i]]
    clxci[, spec := regob[i]]
    Ci0 <- rbind(Ci0, clxci)
  }
  Ci0[, period := gsub("et", "", Coef)]
  Ci0 <- Ci0[grepl("^.?\\d|tre", period), ]
  Ci0[, period := gsub("N", "-", period)]
  Ci0[, period := as.numeric(period)]
  Ci0[, outcome := ob]
  Ci0[, FormulaOrder := factor(FormulaOrder, levels = forder2)]
  setnames(Ci0, c("Estimate", "2.5 %", "97.5 %"), c("beta", "CI_L", "CI_U"))
  Ci <- rbindlist(list(Ci, Ci0), fill = T)
  setcolorder(Ci, c("Coef", "beta", "Std. Error", "t value",
    "Pr(>|t|)", "CI_L", "CI_U", "period"))
  numcols <- c("beta", "CI_L", "CI_U", "period")
  Ci[, (numcols) := lapply(.SD, as.numeric), .SDcols = numcols]
  strcols <- colnames(Ci)[!(colnames(Ci) %in% numcols)]
  Ci[, (strcols) := lapply(.SD, factor), .SDcols = strcols]
}
Ci[grepl("s$", spec) & (abs(period) <= 3 | grepl("trend$", Coef)),
  c("outcome", "Coef", "beta", "Pr(>|t|)", "spec", "FormulaOrder")]
```

```
    outcome  Coef        beta             Pr(>|t|)   spec                   FormulaOrder
 1:      mr trend  0.21817570  9.6624619428244e-89   r32s TWFE itrend premean = 0 N15P10
 2:      mr  etN1  0.09107211    0.743252307349297   r32s TWFE itrend premean = 0 N15P10
 3:      mr  etN3  0.01766871    0.926619850156344   r32s TWFE itrend premean = 0 N15P10
 4:      mr  etN2  0.07287639     0.76784543765025   r32s TWFE itrend premean = 0 N15P10
 5:      mr   et0  0.14294933    0.678906830554418   r32s TWFE itrend premean = 0 N15P10
 6:      mr   et1  0.19091388    0.637144841271113   r32s TWFE itrend premean = 0 N15P10
 7:      mr   et2  0.23560346    0.618110547043362   r32s TWFE itrend premean = 0 N15P10
 8:      mr   et3  0.25970411    0.587705801103398   r32s TWFE itrend premean = 0 N15P10
 9:      mr trend  0.21817570 9.66246194282577e-89 r22ees             TWFE itrend N15P10
10:      mr  etN3 -0.07340340    0.675549267865677 r22ees             TWFE itrend N15P10
11:      mr  etN2 -0.01819572    0.877211590941907 r22ees             TWFE itrend N15P10
12:      mr   et0  0.05187722    0.569473075923662 r22ees             TWFE itrend N15P10
13:      mr   et1  0.09984177     0.50085034303557 r22ees             TWFE itrend N15P10
```

```
14:      mr   et2  0.14453135    0.505841258958236 r22ees             TWFE itrend N15P10
15:      mr   et3  0.16863200    0.486141222286358 r22ees             TWFE itrend N15P10
16:      dv trend  0.08563282 1.07613400606086e-55   r32s TWFE itrend premean = 0 N15P10
17:      dv  etN1 -0.00852053    0.942017103076104   r32s TWFE itrend premean = 0 N15P10
18:      dv  etN3 -0.02653857    0.768006783745344   r32s TWFE itrend premean = 0 N15P10
19:      dv  etN2 -0.06166703    0.586567135171444   r32s TWFE itrend premean = 0 N15P10
20:      dv   et0  0.10585789    0.364559052901692   r32s TWFE itrend premean = 0 N15P10
21:      dv   et1  0.14553214    0.330522283618053   r32s TWFE itrend premean = 0 N15P10
22:      dv   et2  0.16506342    0.360912449172126   r32s TWFE itrend premean = 0 N15P10
23:      dv   et3  0.20476888    0.319709323929253   r32s TWFE itrend premean = 0 N15P10
24:      dv trend  0.08563282 1.07613400606236e-55 r22ees             TWFE itrend N15P10
25:      dv  etN3 -0.01801804    0.875261921679604 r22ees             TWFE itrend N15P10
26:      dv  etN2 -0.05314650    0.592121329751764 r22ees             TWFE itrend N15P10
27:      dv   et0  0.11437842    0.108850954354614 r22ees             TWFE itrend N15P10
28:      dv   et1  0.15405267    0.111409182554798 r22ees             TWFE itrend N15P10
29:      dv   et2  0.17358395    0.120096233778206 r22ees             TWFE itrend N15P10
30:      dv   et3  0.21328941   0.0862050494332137 r22ees             TWFE itrend N15P10
    outcome  Coef        beta              Pr(>|t|)   spec                  FormulaOrder
```

- Under this window restriction, estimates under $\bar{\delta}_{s<0} = 0$ restriction are not diverging.

- Again, in the neighbourhood of when $\delta_t = 0$ is imposed (-10 and -1), SEs are smaller.
- Why are $\bar{R}^2$'s so high?

```
#### Restrict: -15 <= et <= 10
obj = copy(dv)
table(obj[as.numeric(as.character(time)) > 1960, et])
```

```
 -1 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10  -9  -8  -7  -6  -5  -4  -3  -2   0   1
 34   0   0   1   1   2   3   6   9  14  24  32  33  34  34  34  34  34  34  34  34  34  34  34  34  34
  6   7   8   9  10  11  12  13  14  15
 33  33  32  31  28  25  20  10   2   1
```

```
obj1 = copy(dv)
obj1[, time := factor(time,
    levels = c(1988, levels(time)[!(levels(time) %in% c(1988, 1959:1961))], 1961:1959))]
obj1[, et := as.numeric(as.character(et))]
#### et < -15 | et > 10 ==> -1, so it does not directly
#### affect delta (et estimates) once we impose \delta_{-1}=0.
obj1[et < -15 | et > 10, et := -1L]
obj1[, et := factor(et)]
obj1[, et := factor(et, levels = c(-1, levels(et)[!(levels(et) %in% c(-1))]))]
#### r10bb: TWFE with et window restriction
r10bb <- lm(v ~ -1+StateName+time+et,
  data = obj1[as.numeric(as.character(time)) > 1960, ])
#### r22: TWFE with indiv trends and et window restriction
r22 <- lm(v ~ -1+trend*StateName+time+et,
  data = obj1[as.numeric(as.character(time)) > 1960, ])
#### Explicitly drop etN16, etN17, ..., et11, et12, and allow etN1 to be kept
etdumpre <- makeDummyFromFactor(factor(obj[, et]), nameprefix = "et", reference = NULL)
setnames(etdumpre, grepout("-", colnames(etdumpre)),
  gsub("-", "N", grepout("-", colnames(etdumpre))))
#### Drop et < -15 | et > 10 terms
```

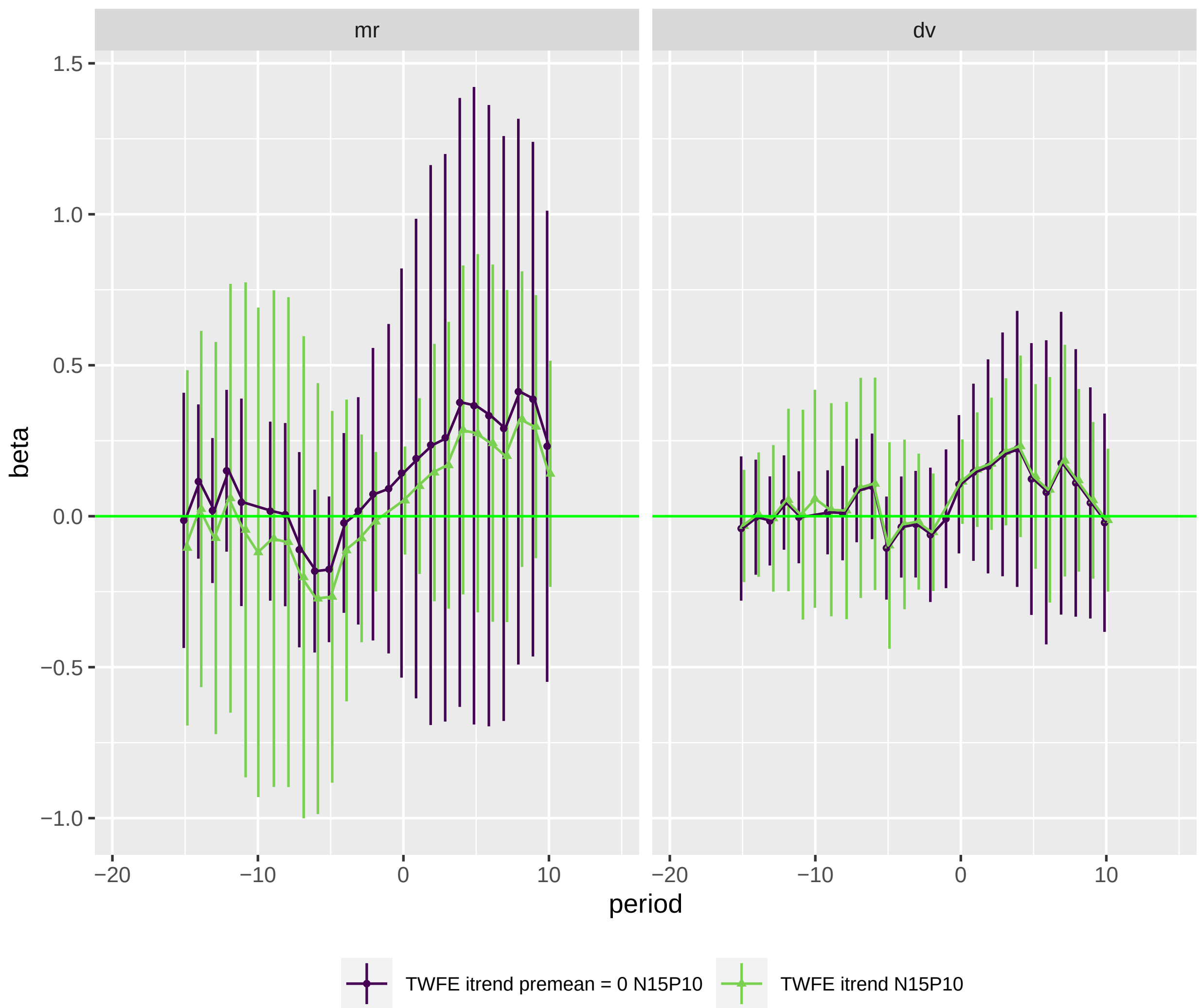

Figure 11: Impacts on divorce and marriage rates: State specific trends, et window N15-10, small sample

```r
etdumpre[, grepout("N2.|N1[6-9]|et1[1-9]", colnames(etdumpre)) := NULL]
ettermspre <- paste(colnames(etdumpre), collapse = "+")
obj3 <- data.table(obj1, etdumpre)
form2 <- paste0("v ~ -1+trend*StateName+time+", ettermspre)
#### r22e: TWFE with indiv trends, et window restriction, keeping N1
#### TWFE itrend N15P10
r22e <- lm(as.formula(form2), data = obj3[as.numeric(as.character(time)) > 1960, ])
#### Explicitly drop et < -15 | et > 10 terms and drop etN1
etdumpre[, etN1 := NULL]
ettermspre <- paste(colnames(etdumpre), collapse = "+")
obj3 <- data.table(obj1, etdumpre)
form2 <- paste0("v ~ -1+trend*StateName+time+", ettermspre)
#### r22ee: TWFE with indiv trends, et window restriction, dropping N1
#### TWFE itrend no N1 N15P10
r22ee <- lm(as.formula(form2), data = obj3[as.numeric(as.character(time)) > 1960, ])
#### Add "TWFE itrend N15P10" to forder, paste N15P10 to r22
forder2 <- c(forder, "TWFE itrend N15P10", "TWFE itrend no N1 N15P10")
forder2 <- gsub("i?trend et pos 2", "itrend et pos 2 N15P10", forder2)
forder2 <- gsub("E et pos 2", "E et pos 2 N15P10", forder2)
regob <- paste0("r", c("10a", "10bb", "10c",
    "22a", "22", "22c", "31", "32", "33", "22e", "22ee"))
Ci <- NULL
for (i in 1:length(regob)) {
  rr <- get(regob[i])
  clus <- data.table(rr$model)[, StateName]
  rrc <- clx(rr, cluster = clus, returnV = T)
  clxci <- data.table(cbind(Coef = rownames(rrc$ci), rrc$est, rrc$ci))
  if (i < 7) clxci <- rbind(clxci, t(c(-1, 0, rep(NA, 5))), use.names = F)
  clxci[, FormulaOrder := forder2[i]]
  clxci[, spec := regob[i]]
  Ci <- rbind(Ci, clxci)
}
Ci[, period := gsub("et", "", Coef)]
Ci <- Ci[grepl("^.?\\d|tre", period), ]
Ci[, period := gsub("N", "-", period)]
Ci[, period := as.numeric(period)]
setcolorder(Ci,
 c("Coef", "Estimate", "Std. Error", "t value", "Pr(>|t|)", "2.5 %", "97.5 %", "period"))
setnames(Ci, c("Estimate", "2.5 %", "97.5 %"), c("beta", "CI_L", "CI_U"))
numcols <- c("beta", "CI_L", "CI_U", "period")
Ci[, (numcols) := lapply(.SD, as.numeric), .SDcols = numcols]
strcols <- colnames(Ci)[!(colnames(Ci) %in% numcols)]
Ci[, (strcols) := lapply(.SD, factor), .SDcols = strcols]
Ci[, FormulaOrder := factor(FormulaOrder, levels = forder2)]
Ci[grepl("10bb|22$|22e|32", spec) & (abs(period) < 2 | grepl("trend$", Coef)),
  c("Coef", "beta", "Pr(>|t|)", "spec", "FormulaOrder")]
```

```
    Coef      beta              Pr(>|t|)  spec                FormulaOrder
 1:   et0 0.0215199     0.847757885171919 r10bb        TWFE et pos 2 N15P10
 2:   et1 0.0602517     0.610317533423476 r10bb        TWFE et pos 2 N15P10
 3:    -1 0.0000000                  <NA> r10bb        TWFE et pos 2 N15P10
 4: trend 0.0856328 1.07613400606236e-55   r22 TWFE itrend et pos 2 N15P10
 5:   et0 0.1143784     0.108850954354614   r22 TWFE itrend et pos 2 N15P10
 6:   et1 0.1540527     0.111409182554798   r22 TWFE itrend et pos 2 N15P10
```

```
 7:    -1   0.0000000                 <NA>   r22 TWFE itrend et pos 2 N15P10
 8: trend   1.5410298  0.00181105356310281   r32      TWFE trend premean = 0
 9:  etN1  -7.5433771  0.00547362374550139   r32      TWFE trend premean = 0
10:   et0  -9.1892336  0.00498351881277007   r32      TWFE trend premean = 0
11:   et1 -10.9991696  0.00445022674543988   r32      TWFE trend premean = 0
12: trend   0.0861365 8.94322972478176e-68  r22e          TWFE itrend N15P10
13:  etN1   0.7175177    0.191440965455812  r22e          TWFE itrend N15P10
14:   et0   0.7734605    0.144947337399478  r22e          TWFE itrend N15P10
15:   et1   0.7897974     0.10840947734539  r22e          TWFE itrend N15P10
16: trend   0.0856328 1.07613400606236e-55 r22ee    TWFE itrend no N1 N15P10
17:   et0   0.1143784    0.108850954354614 r22ee    TWFE itrend no N1 N15P10
18:   et1   0.1540527    0.111409182554798 r22ee    TWFE itrend no N1 N15P10
```

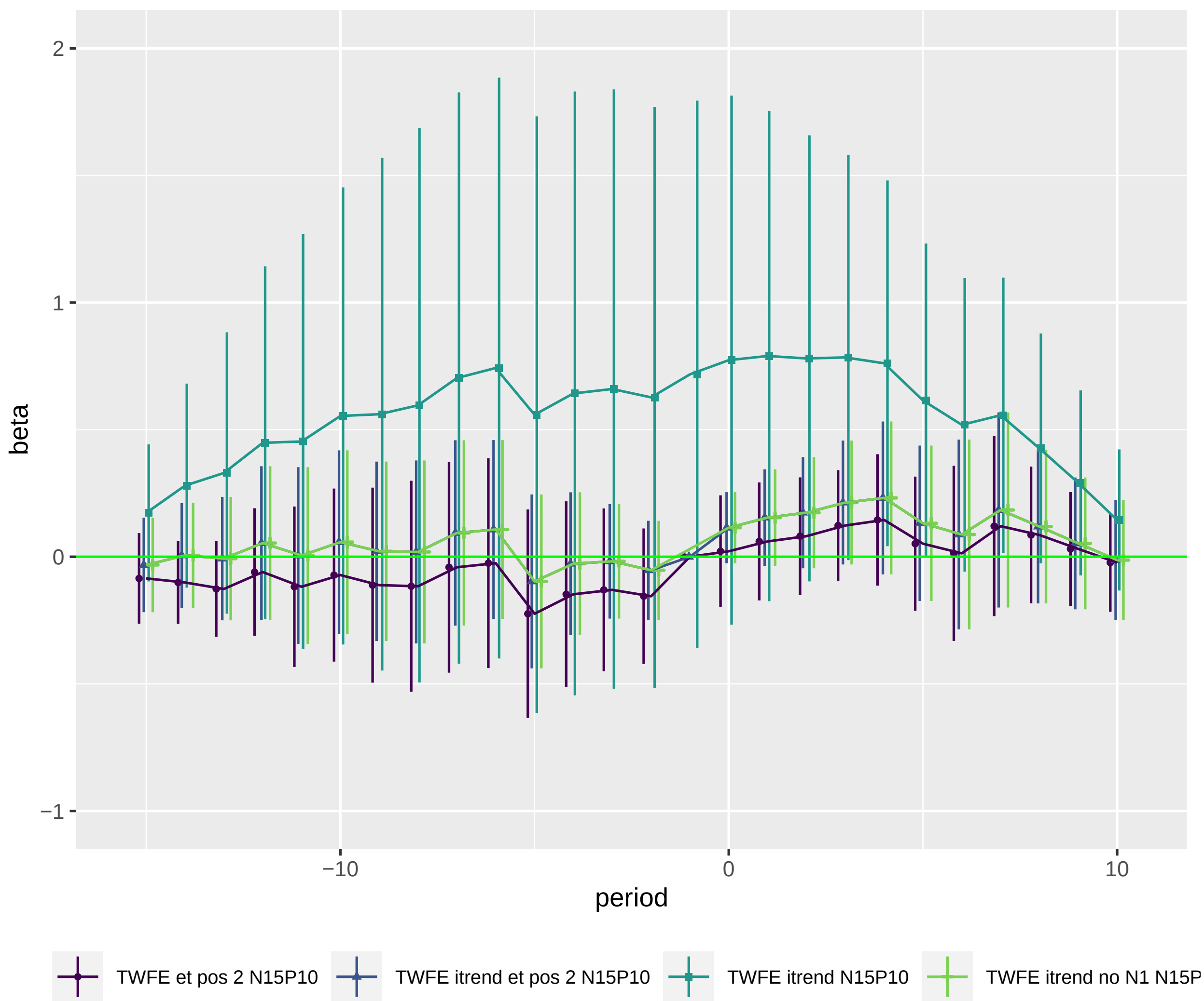

Figure 12: Impacts on divorce rates (event time in -15 to 10, observation 1961 - 1987): With or without state specific trends, keeping or dropping et=-1

- TWFE estimates with state specific trends when we drop et values outside [-15, 10] window may not look similar, but it is all due to the choice of normalisation.

- Dropping et = -1 (TWFE itrend et pos 2 N15P10 estimated using a factor and estimated with an explicit dummy matrix) give identical estimates.
- When not dropping et = -1 (TWFE itrend N15P10), one gets estimates that are pushed upwards as in TWFE itrend N15P10. This is because we are allowing the nonzero estimate at et = -1 ($\delta_{-1} \neq 0$) which shifts the all other $\delta_{t \neq -1}$ estimates to the direction of δ_{-1}. Note the relative size of $\delta_{t \neq -1}$ and δ_{-1} are similar to other normalization choices.

- Not dropping et = -1 (TWFE itrend N15P10) has estimates similar with other normalization choices towards the edge of the window, because these are the furtherst from the difference or et = -1. This effectively gives rise to an inverse-U shape (could have been U shape if the deviation is negative).

- Restricting et = -1 to be zero reduces SEs.

- Estimates with state specific trends have smaller SEs.

- All model estimates are not trending.

- In the base TWFE estimates, States have individual intercepts, year 1987 is dropped from factor `time`, et=-1 is dropped and its estimate is assigned the value of zero in `et`. In TWFE with individual trends, the additional restrictions are: trend*Hawaii is dropped. In TWFE with individual trends and restricting et = -1 to be zero, additional retstriction is et = -1.

So the general idea for normalisation is:

- Manually drop -1 from `et` if normalization $\gamma_{-1} = 0$ is used.

- Manually drop $-L$ from `et` if normalization $\bar{\gamma}_{s<0} = 0$ is used and set $\text{et}_{t<0} = \text{et}_{t<0} - \text{et}_{-L}$.

- Manually drop 2 periods (start and last periods of data) from `time` to incorporate time FE and a linear trend. One for a reference of own dummy variable, another to avoid collinearity with trend.

- One must use an intercept term (or equivalently, use all values of other indicator variable, say, State) to force `et` to be relative to overall mean. Otherwise, it will force $\hat{\delta}_\tau$ to be away from zero when the outcome is trending.

- Manually drop 1 state ("Hawaii") from `StateName` to incorporate state FE with a restriction $\bar{a} = 0$, after setting $a_i = a_i - a_{Hawaii}$.

- Using state specific trends and $\bar{\gamma}_{s<0} = 0$ makes estimates diverge to $-\infty$ or ∞ for reasons I do not understand. Due to unknown reasons, however, setting the event time window narrower as [-15, 10] makes estimates non-divergent.

Veryfying the code with simulated data

Using Bacon data

```
#### https://lost-stats.github.io/Model_Estimation/Research_Design/event_study.html#r
  #### Load and prepare data
####dat = fread("https://raw.githubusercontent.com/LOST-STATS/
#####LOST-STATS.github.io/master/Model_Estimation/Data/Event_Study_DiD/
##### bacon_example.csv")
dat <- fread(FPath("source", "bacon_example.csv"))
  #### Let's create a more user-friendly indicator of which states received treatment
dat[, treat := ifelse(is.na(`_nfd`), 0, 1)]
dat[, time_to_treat := ifelse(treat==1, year - `_nfd`, 0)]
```

`fixest` has `i` function that deals with interaction terms. By default, reference is the first level. Here, -1 is chosen explicitly.

```
library(fixest)
twfe1 = feols(asmrs ~ i(time_to_treat, treat, ref = -1) |
     #### Our key interaction: time × treatment status
     stfips + year,      #### FEs
     cluster = ~stfips, #### Clustered SEs
     data = dat)
```

```
ttdum <- makeDummyFromFactor(factor(dat[, time_to_treat]),
  nameprefix = "tt", reference = NULL)
setnames(ttdum, colnames(ttdum), gsub("-", "N", colnames(ttdum)))
ttterms <- paste(colnames(ttdum), collapse = "+")
ttterms2 <- gsub("\\+ttN1\\+", "+", ttterms) # ttN21+...+ttN2+tt0+tt1+...
ttterms3 <- gsub("ttN21\\+", "", ttterms) # ttN20+...+ttN1+tt0+tt1+...
dat[, time := factor(year)]
dat[, id := factor(stfips)]
dt <- data.table(dat, ttdum)
#### 2: no intercept+ttterms2, et=-1 is dropped
#### 3: with intercept+ttterms2, et=-1, id=1 are dropped
#### 4: with intercept+ttterms3, et=-21, id=1 are dropped
twfe2 <- lm(as.formula(paste0("asmrs ~ -1+id+time+", ttterms2)), data = dt)
twfe3 <- lm(as.formula(paste0("asmrs ~ id+time+", ttterms2)), data = dt)
twfe4 <- lm(as.formula(paste0("asmrs ~ id+time+", ttterms3)), data = dt)
tc1 <- twfe1$coeff
names(tc1) <- gsub("ti.*::", "tt", names(tc1))
names(tc1) <- gsub(":treat", "", names(tc1))
names(tc1) <- gsub("-", "N", names(tc1))
for (i in 1:4) {
  if (i > 1) {
    tw <- get(paste0("twfe", i))
    tc <- tw$coeff
  } else tc <- tc1
  tc <- tc[grep("tt", names(tc))]
  tc <- data.table(spec = c("fixest", "no int, -1", "int, -1, id=1", "int, -21, id=1")[i],
    coef = names(tc), val = tc)
  assign(paste0("tcf", i), tc)
}
tcf <- rbindlist(list(tcf1, tcf2, tcf3, tcf4))
tcf[, et := gsub("tt", "", coef)]
tcf[, et := gsub("N", "-", et)]
tcf[, et := as.numeric(et)]
tcf[et == -21, ]
```

```
            spec  coef      val  et
1:        fixest ttN21 -22.8576 -21
2:    no int, -1 ttN21 -22.8576 -21
3: int, -1, id=1 ttN21 -22.8576 -21
```

```
tcf[et == -1, ]
```

```
             spec coef     val et
1: int, -21, id=1 ttN1 22.8576 -1
```

```r
library(clubSandwich)
ci1 <- data.table(cbind(names(twfe1$coeff), twfe1$coeff, stats::confint(twfe1)))
ci2 <- lapply(list(twfe2, twfe3, twfe4), function(x)
  clubSandwich::conf_int(x, vcov = "CR2", level = 0.95,
  test = "Satterthwaite", cluster = dt[, id], coefs = "All", p_values = T))
ci22 <- lapply(ci2, function(x) data.table(x)[, c("Coef", "beta", "CI_L", "CI_U")])
ci22 <- lapply(ci22, function(x) x[grepl("tt", Coef), ])
ci22 <- lapply(ci22, function(x) x[, et := gsub("tt", "", Coef)])
ci22 <- lapply(ci22, function(x) x[, et := as.numeric(gsub("N", "-", et))])
ci22 <- lapply(1:length(ci22), function(i) ci22[[i]][, estmethod := i+1])
setnames(ci1, c("Coef", "beta", "CI_L", "CI_U"))
ci1[, et := gsub(".*::", "", Coef)]
ci1[, et := as.numeric(gsub(":treat", "", et))]
ci1[, estmethod := 1]
ci22 <- rbindlist(ci22)
ci <- rbindlist(list(ci1, ci22), use.names = T, fill = T)
ci[, estmethod := factor(estmethod, labels = c("fixest", "lm et=-1", "lm et=-1 id=1", "lm et=-21 id=1"))
```

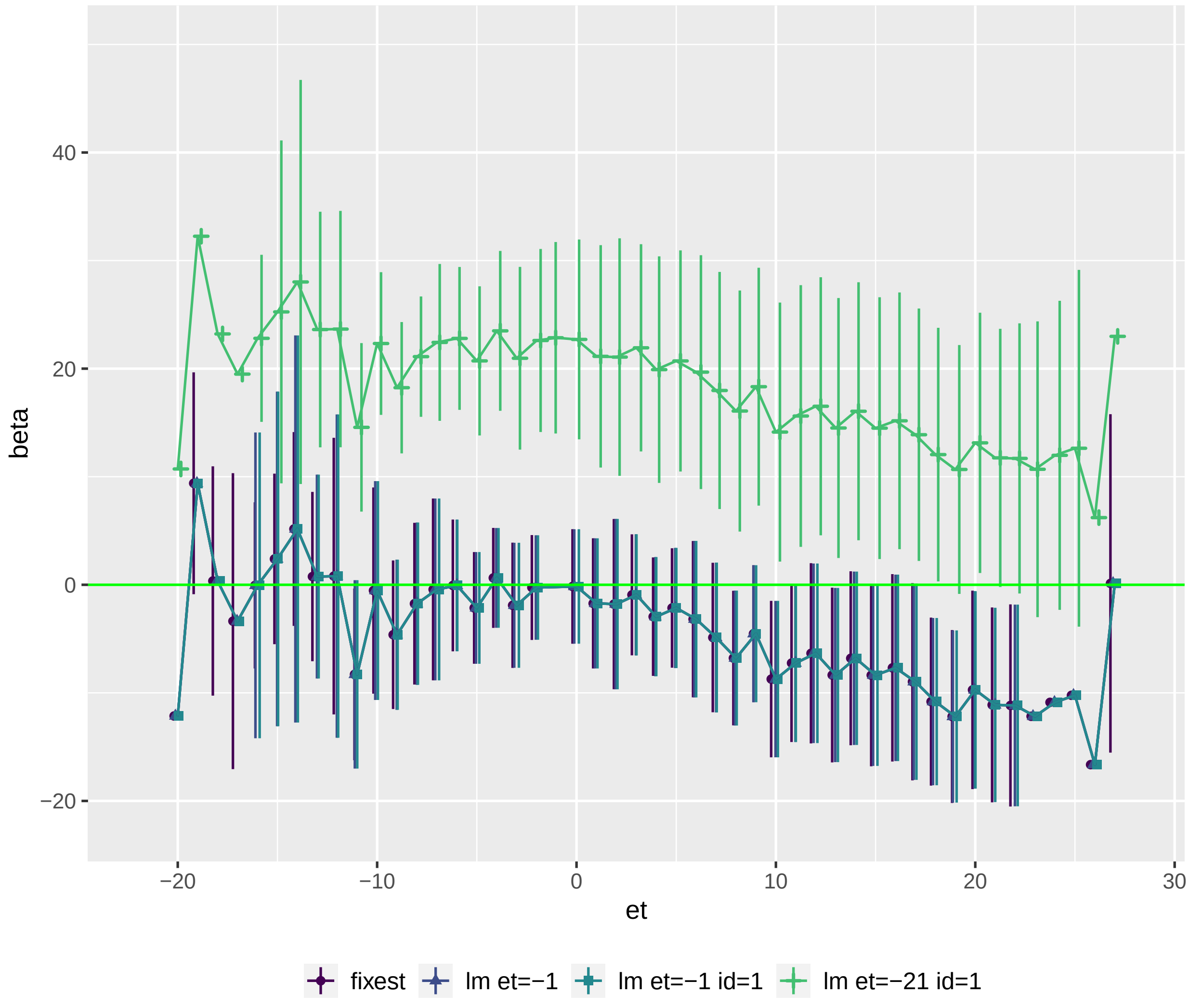

Figure 13: Parameter estimates: ‘fixest‘ and other specifications

We see that `fixest`, "lm et=-1", "lm et=-1 id=1" are equivalent, "lm et=-21" gives the estimates with a pararell shift to the above, and with larger SEs.

Cubic trends

We need to manually drop -1 (for baseline) or $-L = -10$ (for zero mean pre-period effects) from event time variables, Hawaii from State dummy variables (for linear independence), 1960, 1988 from time dummy variables (for accommodating a linear trend). In below, this is done by creating a dummy matrix from a factor variable and dropping the chosen reference. In using restrictions $\bar{\gamma}_{s<0} = 0$ or $\bar{a}_i = 0$, we will subtract the chosen reference from each columns of a dummy matrix.

```
for (ob in c("mr", "dv")) {
  obj <- qread(paste0(pathsave, ob, ".qs"))
  obj <- obj[, time2 := as.numeric(time)]
  obj <- obj[time2 >= 1960 & time2 <= 1988, ]
  #### State dummies
  stdum <- makeDummyFromFactor(factor(obj[, StateName]), nameprefix = "")
    #### Subtract Hawaii to impose \bar{a}_{i} = 0
  stnames <- colnames(stdum)
  setnames(stdum, stnames, gsub(" ", "", stnames))
  stdum[, (stnames) := lapply(.SD, function(x) x-Hawaii), .SDcols = stnames]
  stdum[, Hawaii := NULL]
  stterms <- paste(colnames(stdum), collapse = "+")
  #### Time dummies
  tdum <- makeDummyFromFactor(factor(obj[, time]), nameprefix = "y")
    #### Drop 1961, 1987 for accommodating trend and keep linear independence
  setnames(tdum, colnames(tdum), gsub("19", "", colnames(tdum)))
  tnames <- colnames(tdum)
  tdum[, paste0("y", c(61, 87)) := NULL]
  tterms <- paste(colnames(tdum), collapse = "+")
  etdum <- makeDummyFromFactor(factor(obj[, et]), nameprefix = "et")
  #### Event time dummies
    #### change to easier-to-handle names
  setnames(etdum, grepout("-", colnames(etdum)),
    gsub("-", "N", grepout("-", colnames(etdum))))
  etdumpre = copy(etdum)
  etprepost = copy(etdum)
    #### Subtract t=-L, L=10 period to impose \bar{gamma}_{s<0} = 0
  negtime <- grepout("N", colnames(etdum))
  etdumpre[, (negtime) := lapply(.SD, function(x) x-etN10), .SDcols = negtime]
  etdumpre[, etN10 := NULL]
    #### Subtract t=-1 period to impose
    #### \bar{gamma}_{0} = 0, \gamma_{s\neq 0}=\gamma_{s}-\gamma_{0}
  preposttime <- colnames(etdum)
  etprepost[, (preposttime) := lapply(.SD, function(x) x-etN1),
    .SDcols = preposttime]
  etterms <- paste(colnames(etdum), collapse = "+")
    #### Drop -1 and -10 from et
  etterms1 <- gsub("\\+etN1\\+", "+", etterms)
  etterms2 <- gsub("\\+etN10", "", etterms)
  obj1 <- data.table(obj, stdum, tdum, etdum)
  obj1a <- data.table(obj, stdum, tdum, etprepost)
  obj2 <- data.table(obj, stdum, tdum, etdumpre)
    #### A: TWFE, B: TWFE+trend
```

```r
formA. <- paste0("v ~ -1+", stterms, " + ", tterms)
formB. <- paste0("v ~ -1 + trend +", stterms, " + ", tterms)
formC. <- paste0("v ~ -1 + trend + I(trend^(2)) + I(trend^(3))+",
  stterms, " + ", tterms)
formA1 <- paste(formA., "+", etterms1)
formB1 <- paste(formB., "+", etterms1)
formC1 <- paste(formC., "+", etterms1)
formA2 <- paste(formA., "+", etterms2)
formB2 <- paste(formB., "+", etterms2)
formC2 <- paste(formC., "+", etterms2)
obj1[, id := 1:.N]
obj2[, id := 1:.N]
rA0 <- lm(as.formula(formA.), data = obj1)
rB0 <- lm(as.formula(formB.), data = obj1)
rC0 <- lm(as.formula(formC.), data = obj1)
rA1 <- lm(as.formula(formA1), data = obj1)
rB1 <- lm(as.formula(formB1), data = obj1)
rC1 <- lm(as.formula(formC1), data = obj1)
rA2 <- lm(as.formula(formA2), data = obj2)
rB2 <- lm(as.formula(formB2), data = obj2)
rC2 <- lm(as.formula(formC2), data = obj2)
#### All coefficients are relative to t=-1 (which is set to zero)
rA0a <- lm(as.formula(formA.), data = obj1a)
rB0a <- lm(as.formula(formB.), data = obj1a)
rC0a <- lm(as.formula(formC.), data = obj1a)
rA1a <- lm(as.formula(formA1), data = obj1a)
rB1a <- lm(as.formula(formB1), data = obj1a)
rC1a <- lm(as.formula(formC1), data = obj1a)
assign(paste0(ob, "reg"), list(
  "TWFE"=rA0, "TWFE+t"=rB0, "TWFE+t3"=rC0,
  "TWFE+et"=rA1, "TWFE+t+et"=rB1, "TWFE+t3+et"=rC1,
  "TWFEa"=rA0a, "TWFEa+t"=rB0a, "TWFEa+t3"=rC0a,
  "TWFEa+et"=rA1a, "TWFEa+t+et"=rB1a, "TWFEa+t3+et"=rC1a,
  "TWFE+et, pre-period"=rA2, "TWFE+t+et, pre-period"=rB2,
  "TWFE+t3+et, pre-period"=rC2
  ))
#### CI
normalizationABC <- c("TWFE", "TWFE trend", "TWFE trend3")
normalization123 <- c("no et", "-1", "pre-mean=0")
Ci <- NULL
for (ch in 1:3) {
  for (i in 0:2) {
    for (j in c("", "a")) {
      if (i==2 & j == "a") next
      rr <- get(paste0("r", LETTERS[ch], i, j))
      id <- as.numeric(names(rr$resid))
      clus <- obj1[id, StateName]
      #clus <- data.table(rr$model)[, StateName]
      rrc <- clx(rr, cluster = clus, returnV = T)
      clxci <- data.table(cbind(Coef = rownames(rrc$ci), rrc$est, rrc$ci))
      clxci[, normalABC := gsub("FE", paste0("FE", j), normalizationABC[ch])]
      clxci[, normal123 := normalization123[i+1]]
      Ci <- rbind(Ci, clxci)
```

```
      }
    }
  }
  Ci[, period := gsub("et", "", Coef)]
  Ci <- Ci[grepl("^.?\\d", period), ]
  Ci[, period := gsub("N", "-", period)]
  Ci[, period := as.numeric(period)]
  setcolorder(Ci, c("Coef", "Estimate", "Std. Error", "t value", "Pr(>|t|)",
    "2.5 %", "97.5 %", "period"))
  setnames(Ci, c("Estimate", "2.5 %", "97.5 %"), c("beta", "CI_L", "CI_U"))
  numcols <- c("beta", "CI_L", "CI_U", "period", "Std. Error", "t value", "Pr(>|t|)")
  Ci[, (numcols) := lapply(.SD, as.numeric), .SDcols = numcols]
  strcols <- colnames(Ci)[!(colnames(Ci) %in% numcols)]
  Ci[, (strcols) := lapply(.SD, factor), .SDcols = strcols]
  Ci[grepl("mea", normal123) & period < 0, mean(beta), by = normalABC]
  qsave(Ci, paste0(pathsave, ob, "ci.qs"))
}
```

Trend terms:

```
library(modelsummary)
Results <- list("Divorce rates"=dvreg, "Marriage rates"=mrreg)
ii <- as.vector(which(unlist(lapply(dvreg,
  function(x) any(grepl("tre", names(coef(x))))))))
ii <- ii[ii > 3]
####res <- c(Results[[1]][ii], Results[[2]][ii])
####ms <- modelsummary(res,
####  ####output = "gt",
####  output = "kableExtra",
####  stars = TRUE,
####  title = "Trend terms in two-way FEs of event study estimates",
####  ####coef_omit = "Sta|time|^et.?[123][1-9]|[23]0",
####  #### Need single quotes, double quotes give an error
####  coef_map = c('trend' = 'Linear trend $t$', 'I(trend^(2))' = 'Squared trend $t^{2}$',
####    'I(trend^(3))' = 'Cubic trend $t^{3}$'),
####  gof_omit = 'IC|Adj|F|RMSE|Log')
#### column labels
###library(gt)
###ms <- tab_spanner(data = ms, label = 'Divorce rates', columns = 2:6)
###ms <- tab_spanner(data = ms, label = 'Marriage rates', columns = 7:11)
res <- list("Divorce rates" = Results[[1]][ii], "Marriage rates" = Results[[2]][ii])
ms <- modelsummary(res,
  ####output = "gt",
  output = "kableExtra",
  stars = TRUE,
  shape = "rbind",
  title = "Trend terms in two-way FEs of event study estimates",
  ####coef_omit = "Sta|time|^et.?[123][1-9]|[23]0",
  #### Need single quotes, double quotes give an error
  coef_map = c('trend' = 'Linear trend $t$', 'I(trend^(2))' = 'Squared trend $t^{2}$',
    'I(trend^(3))' = 'Cubic trend $t^{3}$'),
  gof_omit = 'IC|Adj|F|RMSE|Log')
library(kableExtra)
ms <- kable_styling(ms, bootstrap_options = "striped", full_width = F)
```

Table 1: Trend terms in two-way FEs of event study estimates

	TWFE+t+et	TWFE+t3+et	TWFEa+t	TWFEa+t3	TWFEa+t+et	TWFEa+t3+et	TW
Divorce rates							
Linear trend t	0.220***	0.619*	0.177***	1.892***	0.233***	0.625**	
	(0.007)	(0.253)	(0.004)	(0.157)	(0.004)	(0.224)	
Squared trend t^2		-0.121		-0.403***		-0.120	
		(0.078)		(0.063)		(0.073)	
Cubic trend t^3		0.004		0.012***		0.004	
		(0.002)		(0.002)		(0.002)	
Num.Obs.	986	986	986	986	986	986	
R2	0.989	0.989	0.976	0.988	0.989	0.989	
Marriage rates							
Linear trend t	0.495***	4.220***	0.384***	7.393***	0.585***	4.368***	
	(0.017)	(0.549)	(0.014)	(0.338)	(0.010)	(0.488)	
Squared trend t^2		-0.917***		-1.650***		-0.946***	
		(0.169)		(0.136)		(0.160)	
Cubic trend t^3		0.027***		0.048***		0.028***	
		(0.005)		(0.004)		(0.005)	
Num.Obs.	957	957	957	957	957	957	
R2	0.990	0.990	0.951	0.989	0.989	0.990	

Note:
TWFE: two-way fixed effects of years and states; **t**: trend, **et**: even-time dummies; **t3**: linear, squared, a
+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

```
footnote(ms, general = "**TWFE**: two-way fixed effects of years and states; **t**: trend, **et**: even
```

When trending terms (and their cubic terms) are included (and their interactions with state dummies in the richest specifications), event study estimates also tend to have trends similar to gross trends, declining in marriage rates. In the case of divorce rates, use of trend terms breaks down estimation and event study estimates have the magnitude of thousands. This suggests possible collinearity bewteen linear trend and the time-to-event variable. In below, I will use TWFE and + only squared linear trend.

Final remarks

- Choice of normalization can be consequential. A careful choice is needed.
- For the outcomes showing S shaped growth, such as divorce and marriage rates in the US, simple linear trend may not be the best choice.
- Traditionally, statistics used logistic regressions for S shaped outcomes. But this means we cannot use linear models, or TWFE event study design.
- A more flexible trending modelling with TWFE may be fruitful. Cubic trends did not fit well in the current data.

Hansen, Bruce. 2022. *Econometrics*. Princeton University Press.

Miller, Douglas L. 2023. "An Introductory Guide to Event Study Models." *Journal of Economic Perspectives* 37 (2): 203–30. https://doi.org/10.1257/jep.37.2.203.

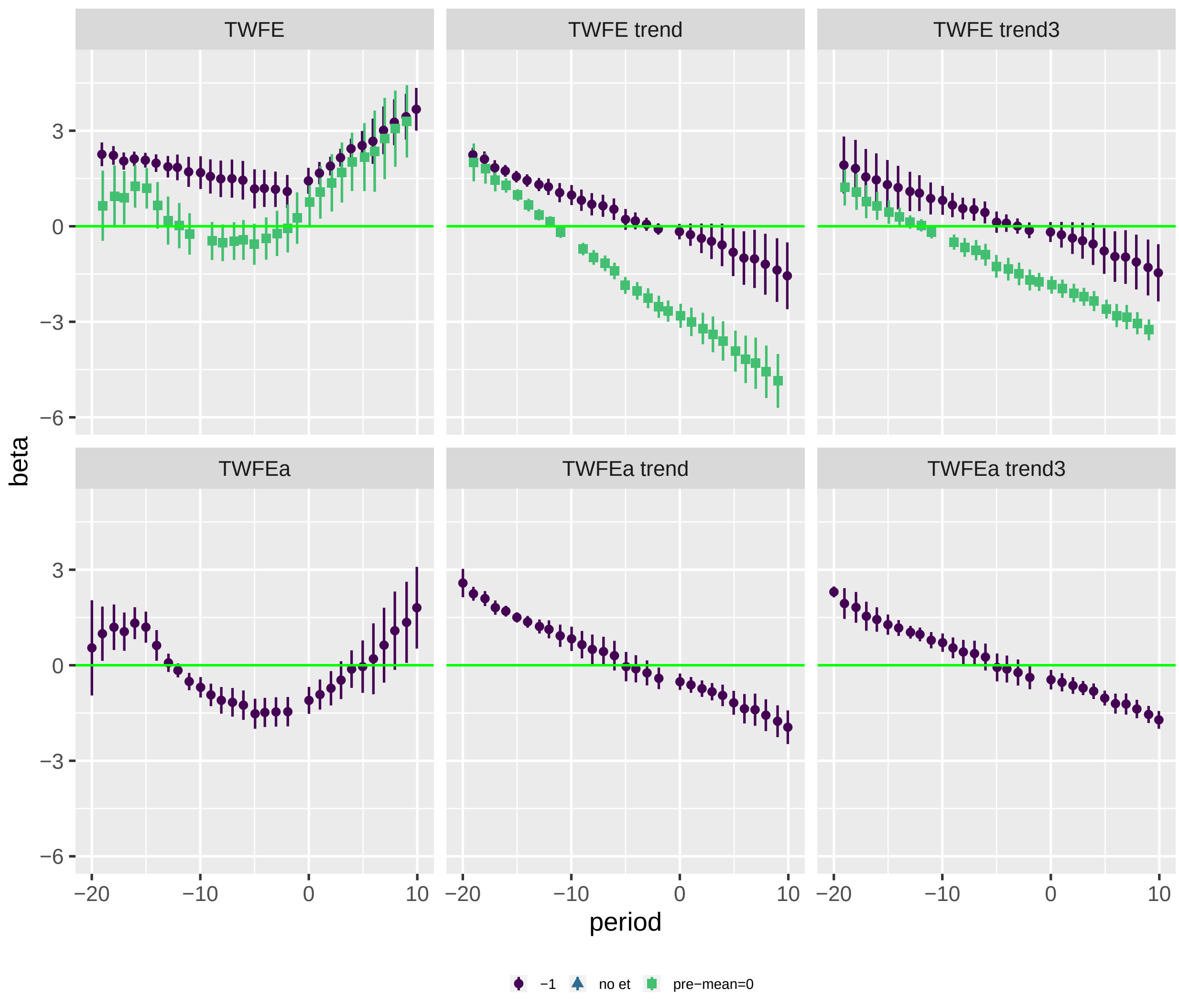

Figure 14: Impacts on divorce rates