```
      name:  <unnamed>
       log:  /Users/connorforsythe/Library/CloudStorage/Box-Box/CMU/Marginal
> Mileage Project/Publication Replication Code/JAERE/PrimaryModel/ReviewRespo
> nseResults/ReviewResponse.smcl
  log type:  smcl
 opened on:   2 Sep 2025, 21:46:56
 1 .         local tableSaveDir = "ReviewResponseResults/"
 2 .         local plotSaveDir = "ReviewResponseResults/"
 3 .         local pythonSaveDir = "PythonScripts/"
 4 . }

 5 .
 6 .
 7 . //Start Timer
 8 . timer on 1

 9 .
10 . //Load useful programs
11 . do programs

12 .
13 . //Adds the first robust first stage f-stat as described in Andrews, Stock,
   > and Sun (2019) p.737; stores result in "fStat"
14 . program addRobustFStat
    1.         version 16.1
    2.         args varToAdd
    3.         estadd scalar fStat = (_b[`varToAdd']^2)/e(V)["`varToAdd'", "`va
   > rToAdd'"]
    4.
15 .
16 . end

17 .
```

```
18 . //Adds the first treatment coefficient and standard error; stores result in
   >  "treatb" and "treatse"
19 . program addTreat
     1.         version 16.1
     2.         args varToAdd
     3.         qui estadd local treatse = "(0"+string(round(_se[`varToAdd'], .0
   > 01))+")"
     4.
20 .
21 .         local tempb = string(round(_b[`varToAdd'], .001))
     5.
22 .         di "hello"
     6.
23 .         local colonIndex = ustrrpos("`varToAdd'", ":")
     7.            local strLen = strlen("`varToAdd'")
     8.            local eqno = udsubstr("`varToAdd'",1,`colonIndex'-1)
     9.            local varName = udsubstr("`varToAdd'",`colonIndex'+1,`strLen'+`c
   > olonIndex'-1)
    10.            di "`colonIndex'"
    11.            di "`eqno'"
    12.            di "`varName'"
    13.
24 .         test [`eqno']`varName'
    14.            di "hello2"
    15.
25 .
26 .         //Determine Stars for alternate null
27 .         local stars = ""
    16.
28 .         if(`r(p)'<0.01){
    17.                    local stars = "***"
    18.            }
    19.            else if(`r(p)'<0.05){
    20.                    local stars = "**"
    21.            }
    22.            else if(`r(p)'<0.1){
    23.                    local stars = "*"
    24.            }
    25.
```

```
29 .         estadd local treatb = "0`tempb'"+"`stars'"
    26.
30 .
31 . end

32 .
33 . //Program to test some specified null;
34 . program testNull, rclass
     1.         version 16.1
     2.         args pi delta piVar deltaVar covar null
     3.
35 .         tempname arStat pVal
     4.
36 .         scalar `arStat' = ((`delta'-`pi'*`null')^2)/(`deltaVar'+(`null')^2*
   > `piVar'-2*`null'*`covar')
     5.         scalar `pVal' = 1-chi2(1, `arStat')
     6.
37 .
38 .         return scalar pi = `pi'
     7.         return scalar delta = `delta'
     8.         return scalar piVar = `piVar'
     9.         return scalar deltaVar = `deltaVar'
    10.         return scalar covar = `covar'
    11.         return scalar arStat = `arStat'
    12.         return scalar pVal = `pVal'
    13.         return scalar null = `null'
    14.         return scalar beta = `delta'/`pi'
    15.
39 . end

40 .
41 . //Program to jointly estimate first stage and reduced form; argument struct
   > ure is first-stage model, reduced form model, cluster variable, treatmentVa
   > r, and the model title
```

```
42 . program jointEstimate, rclass
     1.         version 16.1 //Establish Version
     2.         //Parse arguments
43 .         gettoken fsVars 0:0, parse(",")
     3.         gettoken com 0:0, parse(",")
     4.         gettoken rfVars 0:0, parse(",")
     5.         gettoken com 0:0, parse(",")
     6.         gettoken clusterVar 0:0, parse(",")
     7.         gettoken com 0:0, parse(",")
     8.         gettoken treatmentVar 0:0, parse(",")
     9.         gettoken com 0:0, parse(",")
    10.         gettoken modelTitle 0:0, parse(",")
    11.
44 .         di "Working on:`modelTitle'"
    12.
45 .
46 .
47 .         //Establish Temporary Variables
48 .         tempname fsK rfK N nClust fsCorrection rfCorrection pi delta piVarU
   > ncorrected deltaVarUncorrected covarUncorrected piVar deltaVar covar covarH
   > igh covarLow
    13.
49 .         regress `fsVars'
    14.         scalar `fsK' = e(df_m)+1
    15.         return scalar fsK = `fsK'
    16.         estimates store fs
    17.
50 .         regress `rfVars'
    18.         scalar `rfK' = e(df_m)+1
    19.         return scalar rfK = `rfK'
    20.         estimates store rf
    21.
51 .         qui suest fs rf, vce(cluster `clusterVar')
    22.         scalar `nClust' = e(N_clust)
    23.         scalar `N' = e(N)
    24.         scalar `pi' = e(b)["y1", "fs_mean:`treatmentVar'"]
    25.         scalar `delta' = e(b)["y1", "rf_mean:`treatmentVar'"]
    26.         scalar `piVarUncorrected' = e(V)["fs_mean:`treatmentVar'", "fs_m
   > ean:`treatmentVar'"]
    27.         scalar `deltaVarUncorrected' = e(V)["rf_mean:`treatmentVar'", "r
   > f_mean:`treatmentVar'"]
    28.         scalar `covarUncorrected' = e(V)["rf_mean:`treatmentVar'", "fs_m
   > ean:`treatmentVar'"]
    29.
```

```
52 .
53 .         return scalar nClust = `nClust'
    30.         return scalar N = `N'
    31.         scalar `fsCorrection' = (`N'-1)/(`N'-`fsK'+(`nClust'-1))
    32.         scalar `rfCorrection' = (`N'-1)/(`N'-`rfK'+(`nClust'-1))
    33.         scalar `piVar' = `piVarUncorrected'*`fsCorrection'
    34.         scalar `deltaVar' = `deltaVarUncorrected'*`fsCorrection'
    35.         scalar `covarHigh' = max(`fsCorrection'*`covarUncorrected',`rfCo
   > rrection'*`covarUncorrected')
    36.         scalar `covarLow' = min(`fsCorrection'*`covarUncorrected',`rfCor
   > rection'*`covarUncorrected')
    37.         scalar `covar' = `covarLow'
    38.
54 .         return scalar fsCorrection = `fsCorrection'
    39.         return scalar rfCorrection = `rfCorrection'
    40.         return scalar pi = `pi'
    41.         return scalar delta = `delta'
    42.         return scalar piVarUncorrected = `piVarUncorrected'
    43.         return scalar deltaVarUncorrected = `deltaVarUncorrected'
    44.         return scalar covarUncorrected = `covarUncorrected'
    45.         return scalar piVar = `piVar'
    46.         return scalar deltaVar = `deltaVar'
    47.         return scalar covarHigh = `covarHigh'
    48.         return scalar covarLow = `covarLow'
    49.         return scalar covar = `covar'
    50.
55 .
56 . end

57 .
58 . program regFWL, eclass
     1.         version 16.1 //Establish Version
     2.
59 .         gettoken out_var 0:0, parse(",")
     3.         gettoken com 0:0, parse(",")
     4.         gettoken est_var 0:0, parse(",")
     5.         gettoken com 0:0, parse(",")
     6.         gettoken partial_var 0:0, parse(",")
     7.         gettoken com 0:0, parse(",")
     8.         gettoken absorb_vars 0:0, parse(",")
     9.         gettoken com 0:0, parse(",")
    10.         gettoken cluster_var 0:0, parse(",")
    11.         gettoken com 0:0, parse(",")
    12.         gettoken res_var 0:0, parse(",")
    13.         gettoken com 0:0, parse(",")
    14.
```

```
60 .
61 .         if("`out_var'"=="_") local out_var = ""
    15.         if("`est_var'"=="_") local est_var = ""
    16.         if("`partial_var'"=="_") local partial_var = ""
    17.         if("`absorb_vars'"=="_") local absorb_vars = ""
    18.         if("`cluster_var'"=="_") local cluster_var = ""
    19.         if("`res_var'"=="_") local res_var = ""
    20.
62 .         local numClusterVar = 0
    21.
63 .         foreach tempVar in `cluster_var'{
    22.                 local numClusterVar = `numClusterVar'+1
    23.                 local clusterVar`numClusterVar' = "`tempVar'"
    24.         }
    25.         di "absorb: `absorb_vars'"
    26.         qui reghdfe `out_var' `partial_var', absorb(`absorb_vars') res(`
   > out_var'_res)
    27.         local new_dep = "`out_var'_res"
    28.         local new_ind = ""
    29.
64 .         local num_est = 0
    30.         if("`est_var'"!=""){
    31.                 foreach temp_var in `est_var'{
    32.                         qui reghdfe `temp_var' `partial_var', absorb(`ab
   > sorb_vars') res(`temp_var'_res)
    33.                         local new_ind = "`new_ind' `temp_var'_res"
    34.                         local num_est = `num_est'+1
    35.                 }
    36.         }
    37.
65 .         //Run just to get correct values for k; this is the full regression
   > , so other statistics that are unrelated to covariance can be used
66 .         if("`res_var'"==""){
    38.                 qui reghdfe `out_var' `est_var' `partial_var', absorb(`a
   > bsorb_vars') cluster(`cluster_var')
    39.         }
    40.         else{
    41.                 qui reghdfe `out_var' `est_var' `partial_var', absorb(`a
   > bsorb_vars') cluster(`cluster_var') res(`res_var')
    42.         }
    43.         local k = e(df_a)+e(df_m)
    44.         local N = e(N)
    45.         local r2 = e(r2)
    46.         qui reghdfe `new_dep' `new_ind', noabsorb nocon cluster(`cluster
   > _var')
    47.
```

```
67 .         qui regress `new_dep' `new_ind', nocon
    48.         ereturn list
    49.
68 .         estimates store mod1
    50.
69 .         qui suest mod1, vce(cluster `clusterVar1')
    51.         local nClust1 = e(N_clust)
    52.         mat var1 = e(V)
    53.
70 .         qui suest mod1, vce(cluster `clusterVar2')
    54.         local nClust2 = e(N_clust)
    55.         mat var2 = e(V)
    56.
71 .         qui suest mod1, vce(robust)
    57.
72 .         mat varR = e(V)
    58.         mat beta = e(b)
    59.
73 .
74 .         mat mat_var = var1+var2-varR
    60.         local nClust = min(`nClust1', `nClust2')
    61.         local correction = (`nClust')/(`nClust'-1)*(`N'-1)/(`N'-`k')
    62.         mat mat_var = `correction'*mat_var
    63.         matlist mat_var
    64.         ereturn post beta mat_var
    65.         ereturn scalar r2 = `r2'
    66.         ereturn scalar N = `N'
    67.         ereturn scalar N_clust = `nClust'
    68.         drop `new_dep' `new_ind'
    69.
75 .         //Complete
76 . end

77 .
```

```
78 .
79 .
80 . program jointEstimateFWL, rclass
     1.         version 16.1 //Establish Version
     2.         //Parse arguments
81 .         gettoken fsOut 0:0, parse(",")
     3.         gettoken com 0:0, parse(",")
     4.         gettoken rfOut 0:0, parse(",")
     5.         gettoken com 0:0, parse(",")
     6.         gettoken fsControls 0:0, parse(",")
     7.         gettoken com 0:0, parse(",")
     8.         gettoken rfControls 0:0, parse(",")
     9.         gettoken com 0:0, parse(",")
    10.         gettoken treatControl 0:0, parse(",")
    11.         gettoken com 0:0, parse(",")
    12.         gettoken absorbVars 0:0, parse(",")
    13.         gettoken com 0:0, parse(",")
    14.         gettoken clusterVar 0:0, parse(",")
    15.         gettoken com 0:0, parse(",")
    16.         gettoken treatmentVar 0:0, parse(",")
    17.         gettoken com 0:0, parse(",")
    18.         gettoken modelTitle 0:0, parse(",")
    19.
82 .         di "Working on:`modelTitle'"
    20.
83 .         local numClusterVar = 0
    21.
84 .         foreach tempVar in `clusterVar'{
    22.                 local numClusterVar = `numClusterVar'+1
    23.                 local clusterVar`numClusterVar' = "`tempVar'"
    24.         }
    25.
85 .         //Establish Temporary Variables
86 .         tempname fsK rfK N nClust nClust1 nClust2 fsCorrection rfCorrection
   >  pi delta piVarUncorrected deltaVarUncorrected covarUncorrected piVar delta
   > Var covar covarHigh covarLow piVarUncorrected1 deltaVarUncorrected1 covarUn
   > corrected1 piVarUncorrected2 deltaVarUncorrected2 covarUncorrected2 piVarUn
   > correctedR deltaVarUncorrectedR covarUncorrectedR
    26.
```

```
87 .         qui reghdfe `fsOut' `fsControls', absorb(`absorbVars') res(`fsOut'_
   > res) cluster(`clusterVar')
 27.         scalar `fsK' = e(df_a)+e(df_m)+1 //Adding 1 because second regre
   > ssion only has one regressor by construction, so this accounts for all regr
   > essors
 28.
88 .         qui reghdfe `treatControl' `fsControls', absorb(`absorbVars') res(f
   > s_`treatControl'_res)
 29.
89 .         regress `fsOut'_res fs_`treatControl'_res, nocon
 30.
90 .         return scalar fsK = `fsK'
 31.         estimates store fs
 32.
91 .         qui reghdfe `rfOut' `rfControls', absorb(`absorbVars') res(`rfOut'_
   > res) cluster(`clusterVar')
 33.         scalar `rfK' = e(df_a)+e(df_m)+1 //Adding 1 because second regre
   > ssion only has one regressor by construction, so this accounts for all regr
   > essors
 34.
92 .         qui reghdfe `treatControl' `rfControls', absorb(`absorbVars') res(r
   > f_`treatControl'_res)
 35.
93 .         qui regress `rfOut'_res rf_`treatControl'_res, nocon
 36.
94 .
95 .         return scalar rfK = `rfK'
 37.         estimates store rf
 38.
96 .         if `numClusterVar'==1{
 39.                 qui suest fs rf, vce(cluster `clusterVar')
 40.
97 .                scalar `nClust' = e(N_clust)
 41.                scalar `N' = e(N)
 42.                scalar `pi' = e(b)["y1", "fs_mean:fs_`treatControl'_res"
   > ]
 43.                scalar `delta' = e(b)["y1", "rf_mean:rf_`treatControl'_r
   > es"]
 44.                scalar `piVarUncorrected' = e(V)["fs_mean:fs_`treatContr
   > ol'_res", "fs_mean:fs_`treatControl'_res"]
 45.                scalar `deltaVarUncorrected' = e(V)["rf_mean:rf_`treatCo
   > ntrol'_res", "rf_mean:rf_`treatControl'_res"]
 46.                scalar `covarUncorrected' = e(V)["rf_mean:rf_`treatContr
   > ol'_res", "fs_mean:fs_`treatControl'_res"]
 47.         }
 48.         else{
```

```
 49.                 di "Current approach can handle two-way clustering. High
> er dimensional clustering will produce incorrect results."
 50.                 qui suest fs rf, vce(cluster `clusterVar1')
 51.                 scalar `nClust1' = e(N_clust)
 52.                 scalar `N' = e(N)
 53.                 scalar `pi' = e(b)["y1", "fs_mean:fs_`treatControl'_res"
> ]
 54.                 scalar `delta' = e(b)["y1", "rf_mean:rf_`treatControl'_r
> es"]
 55.                 scalar `piVarUncorrected1' = e(V)["fs_mean:fs_`treatCont
> rol'_res", "fs_mean:fs_`treatControl'_res"]
 56.                 scalar `deltaVarUncorrected1' = e(V)["rf_mean:rf_`treatC
> ontrol'_res", "rf_mean:rf_`treatControl'_res"]
 57.                 scalar `covarUncorrected1' = e(V)["rf_mean:rf_`treatCont
> rol'_res", "fs_mean:fs_`treatControl'_res"]
 58.
 98 .                 return scalar piVarUncorrected1 = `piVarUncorrected1'
 59.                 return scalar deltaVarUncorrected1 = `deltaVarUncorrecte
> d1'
 60.                 return scalar covarUncorrected1 = `covarUncorrected1'
 61.
 99 .                 qui suest fs rf, vce(cluster `clusterVar2')
 62.
100 .                 scalar `nClust2' = e(N_clust)
 63.                 scalar `piVarUncorrected2' = e(V)["fs_mean:fs_`treatCont
> rol'_res", "fs_mean:fs_`treatControl'_res"]
 64.                 scalar `deltaVarUncorrected2' = e(V)["rf_mean:rf_`treatC
> ontrol'_res", "rf_mean:rf_`treatControl'_res"]
 65.                 scalar `covarUncorrected2' = e(V)["rf_mean:rf_`treatCont
> rol'_res", "fs_mean:fs_`treatControl'_res"]
 66.
101 .                 return scalar piVarUncorrected2 = `piVarUncorrected2'
 67.                 return scalar deltaVarUncorrected2 = `deltaVarUncorrecte
> d2'
 68.                 return scalar covarUncorrected2 = `covarUncorrected2'
 69.
```

```
102 .                 qui suest fs rf, vce(robust)
     70.
103 .                 scalar `piVarUncorrectedR' = e(V)["fs_mean:fs_`treatControl
    > '_res", "fs_mean:fs_`treatControl'_res"]
     71.                 scalar `deltaVarUncorrectedR' = e(V)["rf_mean:rf_`treatC
    > ontrol'_res", "rf_mean:rf_`treatControl'_res"]
     72.                 scalar `covarUncorrectedR' = e(V)["rf_mean:rf_`treatCont
    > rol'_res", "fs_mean:fs_`treatControl'_res"]
     73.
104 .                 return scalar piVarUncorrectedR = `piVarUncorrectedR'
     74.                 return scalar deltaVarUncorrectedR = `deltaVarUncorrecte
    > dR'
     75.                 return scalar covarUncorrectedR = `covarUncorrectedR'
     76.
105 .                 scalar `nClust' = min(`nClust1', `nClust2')
     77.                 return scalar nClust1 = `nClust1'
     78.                 return scalar nClust2 = `nClust2'
     79.                 scalar `piVarUncorrected' = `piVarUncorrected1'+`piVarUn
    > corrected2'-`piVarUncorrectedR'
     80.                 scalar `deltaVarUncorrected' = `deltaVarUncorrected1'+`d
    > eltaVarUncorrected2'-`deltaVarUncorrectedR'
     81.                 scalar `covarUncorrected' = `covarUncorrected1'+`covarUn
    > corrected2'-`covarUncorrectedR'
     82.         }
     83.
106 .
107 .
108 .         return scalar nClust = `nClust'
     84.         return scalar N = `N'
     85.         scalar `fsCorrection' = (`nClust')/(`nClust'-1)*(`N'-1)/(`N'-`fs
    > K')
     86.         scalar `rfCorrection' = (`nClust')/(`nClust'-1)*(`N'-1)/(`N'-`rf
    > K')
     87.         scalar `piVar' = `piVarUncorrected'*`fsCorrection'
     88.         scalar `deltaVar' = `deltaVarUncorrected'*`fsCorrection'
     89.         scalar `covarHigh' = max(`fsCorrection'*`covarUncorrected',`rfCo
    > rrection'*`covarUncorrected')
     90.         scalar `covarLow' = min(`fsCorrection'*`covarUncorrected',`rfCor
    > rection'*`covarUncorrected')
     91.         scalar `covar' = `covarLow'
     92.
```

```
109 .         return scalar fsCorrection = `fsCorrection'
     93.         return scalar rfCorrection = `rfCorrection'
     94.         return scalar pi = `pi'
     95.         return scalar delta = `delta'
     96.         return scalar piVarUncorrected = `piVarUncorrected'
     97.         return scalar deltaVarUncorrected = `deltaVarUncorrected'
     98.         return scalar covarUncorrected = `covarUncorrected'
     99.         return scalar piVar = `piVar'
    100.         return scalar deltaVar = `deltaVar'
    101.         return scalar covarHigh = `covarHigh'
    102.         return scalar covarLow = `covarLow'
    103.         return scalar covar = `covar'
    104.
110 .         drop `fsOut'_res fs_`treatControl'_res `rfOut'_res rf_`treatControl
    > '_res
    105.
111 . end

112 .
113 .
114 . //Program to output jointly estimated resuls along with a range of null hyp
    > otheses; argument structure is first-stage model, reduced form model, clust
    > er variable, treatmentVar, modelTitle, nullLow, nullDelta, nullHigh, primar
    > y null
115 . program runModel
      1.         version 16.1
      2.         gettoken fsVars 0:0, parse(",")
      3.         gettoken com 0:0, parse(",")
      4.         gettoken rfVars 0:0, parse(",")
      5.         gettoken com 0:0, parse(",")
      6.         gettoken clusterVar 0:0, parse(",")
      7.         gettoken com 0:0, parse(",")
      8.         gettoken treatmentVar 0:0, parse(",")
      9.         gettoken com 0:0, parse(",")
     10.         gettoken modelTitle 0:0, parse(",")
     11.         gettoken com 0:0, parse(",")
     12.         gettoken nullLow 0:0, parse(",")
     13.         gettoken com 0:0, parse(",")
     14.         gettoken nullDelta 0:0, parse(",")
     15.         gettoken com 0:0, parse(",")
     16.         gettoken nullHigh 0:0, parse(",")
     17.         gettoken com 0:0, parse(",")
     18.         gettoken primaryNull 0:0, parse(",")
     19.         gettoken com 0:0, parse(",")
     20.         gettoken altNull 0:0, parse(",")
     21.         gettoken com 0:0, parse(",")
```

```
 22.         gettoken altNull2 0:0, parse(",")
 23.         gettoken com 0:0, parse(",")
 24.         gettoken altNull3 0:0, parse(",")
 25.         gettoken com 0:0, parse(",")
 26.         gettoken altNull4 0:0, parse(",")
 27.
116 .         tempname null tempPi tempDelta tempPiVar tempDeltaVar tempCovar fil
    > eName
 28.
117 .
118 .         if("`c(os)'"=="Windows"){
 29.                 local tableSaveDir = "Tables\"
 30.                 local plotSaveDir = "Plots\"
 31.                 local pythonSaveDir = "PythonScripts\"
 32.         }
 33.         else{
 34.                 local tableSaveDir = "Tables/"
 35.                 local plotSaveDir = "Plots/"
 36.                 local pythonSaveDir = "PythonScripts/"
 37.         }
 38.
119 .         local fileName = "`tableSaveDir'iv-"+"`modelTitle'"+".xlsx"
 39.
120 .         jointEstimate `fsVars', `rfVars', `clusterVar', `treatmentVar', `mo
    > delTitle'
 40.         return list
 41.
121 .         scalar `tempPi' = `r(pi)'
 42.         scalar `tempDelta' = `r(delta)'
 43.         scalar `tempPiVar' = `r(piVar)'
 44.         scalar `tempDeltaVar' = `r(deltaVar)'
 45.         scalar `tempCovar' = `r(covar)'
 46.
122 .         python script "`pythonSaveDir'calculateARStat.py", args(`tempPi' `t
    > empDelta' `tempPiVar' `tempDeltaVar' `tempCovar' `nullLow' `nullDelta' `nul
    > lHigh' `modelTitle')
 47.
```

```
123 .         putexcel set "`fileName'", sheet("JointResults") modify
     48.
124 .         local primaryNullString = string(`primaryNull', "%2.1f")
     49.           local altNullString = string(`altNull', "%2.1f")
     50.
125 .         qui putexcel B1 = `r(pi)'
     51.           qui putexcel B2 = `r(delta)'
     52.           qui putexcel B3 = `r(piVar)'
     53.           qui putexcel B4 = `r(deltaVar)'
     54.           qui putexcel B5 = `r(covar)'
     55.           qui putexcel B6 = `r(covarHigh)'
     56.           qui putexcel B7 = `r(covarLow)'
     57.           qui putexcel B8 = `r(piVarUncorrected)'
     58.           qui putexcel B9 = `r(deltaVarUncorrected)'
     59.           qui putexcel B10 = `r(covarUncorrected)'
     60.           qui putexcel B11 = `r(fsCorrection)'
     61.           qui putexcel B12 = `r(rfCorrection)'
     62.           qui putexcel B13 = `r(N)'
     63.           qui putexcel B14 = `r(nClust)'
     64.           qui putexcel B15 = `r(fsK)'
     65.           qui putexcel B16 = `r(rfK)'
     66.
126 .         qui putexcel A1 = "pi"
     67.           qui putexcel A2 = "delta"
     68.           qui putexcel A3 = "pi variance"
     69.           qui putexcel A4 = "delta variance"
     70.           qui putexcel A5 = "pi-delta covariance"
     71.           qui putexcel A6 = "pi-delta covariance high"
     72.           qui putexcel A7 = "pi-delta covariance low"
     73.           qui putexcel A8 = "pi variance uncorrected"
     74.           qui putexcel A9 = "delta variance uncorrected"
     75.           qui putexcel A10 = "pi-delta covariance uncorrected"
     76.           qui putexcel A11 = "first stage finite-sample correction"
     77.           qui putexcel A12 = "reduced form finite-sample correction"
     78.           qui putexcel A13 = "N"
     79.           qui putexcel A14 = "N Clusters"
     80.           qui putexcel A15 = "first stage k"
     81.           qui putexcel A16 = "reduced form k"
     82.           qui putexcel A17 = "AR(Beta=`primaryNullString')"
     83.           qui putexcel A18 = "P-Value(Beta=`primaryNullString')"
     84.           qui putexcel A19 = "beta"
     85.           qui putexcel A20 = "AR(Beta=`altNullString')"
     86.           qui putexcel A21 = "P-Value(Beta=`altNullString')"
     87.           qui putexcel A22 = "AR(Beta=`altNull2')"
     88.           qui putexcel A23 = "P-Value(Beta=`altNull2')"
     89.           qui putexcel A24 = "AR(Beta=`altNull3')"
```

```
 90.         qui putexcel A25 = "P-Value(Beta=`altNull3')"
 91.         qui putexcel A26 = "AR(Beta=`altNull4')"
 92.         qui putexcel A27 = "P-Value(Beta=`altNull4')"
 93.
127 .
128 .
129 .         testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `tempCovar
    > ' `primaryNull'
 94.
130 .         qui putexcel B17 = `r(arStat)'
 95.         qui putexcel B18 = `r(pVal)'
 96.         qui putexcel B19 = `r(beta)'
 97.
131 .         testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `tempCovar
    > ' `altNull'
 98.         qui putexcel B20 = `r(arStat)'
 99.         qui putexcel B21 = `r(pVal)'
100.
132 .         testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `tempCovar
    > ' `altNull2'
101.         qui putexcel B22 = `r(arStat)'
102.         qui putexcel B23 = `r(pVal)'
103.
133 .         if("`altNull2'"!="`altNull3'"){
104.                 qui putexcel A24 = "AR(Beta=`altNull3')"
105.                 qui putexcel A25 = "P-Value(Beta=`altNull3')"
106.
134 .                 testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `t
    > empCovar' `altNull3'
107.                 qui putexcel B24 = `r(arStat)'
108.                 qui putexcel B25 = `r(pVal)'
109.         }
110.         if("`altNull3'"!="`altNull4'"){
111.                 qui putexcel A26 = "AR(Beta=`altNull4')"
112.                 qui putexcel A27 = "P-Value(Beta=`altNull4')"
113.
```

```
135 .                 testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `t
    > empCovar' `altNull4'
    114.                 qui putexcel B26 = `r(arStat)'
    115.                 qui putexcel B27 = `r(pVal)'
    116.         }
    117.
136 .
137 . end

138 .
139 . //Program to output jointly estimated resuls along with a range of null hyp
    > otheses; argument structure is first-stage model, reduced form model, clust
    > er variable, treatmentVar, modelTitle, nullLow, nullDelta, nullHigh, primar
    > y null
140 . program runModelFWL
      1.         version 16.1
      2.         gettoken fsOut 0:0, parse(",")
      3.         gettoken com 0:0, parse(",")
      4.         gettoken rfOut 0:0, parse(",")
      5.         gettoken com 0:0, parse(",")
      6.         gettoken fsControls 0:0, parse(",")
      7.         gettoken com 0:0, parse(",")
      8.         gettoken rfControls 0:0, parse(",")
      9.         gettoken com 0:0, parse(",")
     10.         gettoken treatControl 0:0, parse(",")
     11.         gettoken com 0:0, parse(",")
     12.         gettoken absorbVars 0:0, parse(",")
     13.         gettoken com 0:0, parse(",")
     14.         gettoken clusterVar 0:0, parse(",")
     15.         gettoken com 0:0, parse(",")
     16.         gettoken treatmentVar 0:0, parse(",")
     17.         gettoken com 0:0, parse(",")
     18.         gettoken modelTitle 0:0, parse(",")
     19.         gettoken com 0:0, parse(",")
     20.         gettoken nullLow 0:0, parse(",")
     21.         gettoken com 0:0, parse(",")
     22.         gettoken nullDelta 0:0, parse(",")
     23.         gettoken com 0:0, parse(",")
     24.         gettoken nullHigh 0:0, parse(",")
     25.         gettoken com 0:0, parse(",")
     26.         gettoken primaryNull 0:0, parse(",")
     27.         gettoken com 0:0, parse(",")
     28.         gettoken altNull 0:0, parse(",")
     29.         gettoken com 0:0, parse(",")
     30.         gettoken altNull2 0:0, parse(",")
     31.         gettoken com 0:0, parse(",")
```

```
     32.         gettoken altNull3 0:0, parse(",")
     33.         gettoken com 0:0, parse(",")
     34.         gettoken altNull4 0:0, parse(",")
     35.
141 .         tempname null tempPi tempDelta tempPiVar tempDeltaVar tempCovar fil
    > eName
     36.
142 .
143 .         if("`c(os)'"=="Windows"){
     37.                 local tableSaveDir = "Tables\"
     38.                 local plotSaveDir = "Plots\"
     39.                 local pythonSaveDir = "PythonScripts\"
     40.         }
     41.         else{
     42.                 local tableSaveDir = "Tables/"
     43.                 local plotSaveDir = "Plots/"
     44.                 local pythonSaveDir = "PythonScripts/"
     45.         }
     46.
144 .         local fileName = "`tableSaveDir'iv-"+"`modelTitle'"+".xlsx"
     47.
145 .         di "`fsOut'"
     48.         di "`rfOut'"
     49.         di "`fsControls'"
     50.         di "`rfControls'"
     51.         di "`treatControl'"
     52.         di "`absorbVars'"
     53.
146 .         jointEstimateFWL `fsOut', `rfOut', `fsControls', `rfControls', `tre
    > atControl', `absorbVars', `clusterVar', `treatmentVar', `modelTitle'
     54.         return list
     55.
147 .         scalar `tempPi' = `r(pi)'
     56.         scalar `tempDelta' = `r(delta)'
     57.         scalar `tempPiVar' = `r(piVar)'
     58.         scalar `tempDeltaVar' = `r(deltaVar)'
     59.         scalar `tempCovar' = `r(covar)'
     60.
```

```
148 .         python script "`pythonSaveDir'calculateARStat.py", args(`tempPi' `t
    > empDelta' `tempPiVar' `tempDeltaVar' `tempCovar' `nullLow' `nullDelta' `nul
    > lHigh' `modelTitle')
     61.
149 .         putexcel set "`fileName'", sheet("JointResults") modify
     62.
150 .         local primaryNullString = string(`primaryNull', "%2.1f")
     63.            local altNullString = string(`altNull', "%2.1f")
     64.
151 .         qui putexcel B1 = `r(pi)'
     65.            qui putexcel B2 = `r(delta)'
     66.            qui putexcel B3 = `r(piVar)'
     67.            qui putexcel B4 = `r(deltaVar)'
     68.            qui putexcel B5 = `r(covar)'
     69.            qui putexcel B6 = `r(covarHigh)'
     70.            qui putexcel B7 = `r(covarLow)'
     71.            qui putexcel B8 = `r(piVarUncorrected)'
     72.            qui putexcel B9 = `r(deltaVarUncorrected)'
     73.            qui putexcel B10 = `r(covarUncorrected)'
     74.            qui putexcel B11 = `r(fsCorrection)'
     75.            qui putexcel B12 = `r(rfCorrection)'
     76.            qui putexcel B13 = `r(N)'
     77.            qui putexcel B14 = `r(nClust)'
     78.            qui putexcel B15 = `r(fsK)'
     79.            qui putexcel B16 = `r(rfK)'
     80.
152 .         qui putexcel A1 = "pi"
     81.            qui putexcel A2 = "delta"
     82.            qui putexcel A3 = "pi variance"
     83.            qui putexcel A4 = "delta variance"
     84.            qui putexcel A5 = "pi-delta covariance"
     85.            qui putexcel A6 = "pi-delta covariance high"
     86.            qui putexcel A7 = "pi-delta covariance low"
     87.            qui putexcel A8 = "pi variance uncorrected"
     88.            qui putexcel A9 = "delta variance uncorrected"
     89.            qui putexcel A10 = "pi-delta covariance uncorrected"
     90.            qui putexcel A11 = "first stage finite-sample correction"
     91.            qui putexcel A12 = "reduced form finite-sample correction"
     92.            qui putexcel A13 = "N"
     93.            qui putexcel A14 = "N Clusters"
     94.            qui putexcel A15 = "first stage k"
     95.            qui putexcel A16 = "reduced form k"
     96.            qui putexcel A17 = "AR(Beta=`primaryNullString')"
     97.            qui putexcel A18 = "P-Value(Beta=`primaryNullString')"
     98.            qui putexcel A19 = "beta"
     99.            qui putexcel A20 = "AR(Beta=`altNullString')"
```

```
100.         qui putexcel A21 = "P-Value(Beta=`altNullString')"
101.         qui putexcel A22 = "AR(Beta=`altNull2')"
102.         qui putexcel A23 = "P-Value(Beta=`altNull2')"
103.         qui putexcel A24 = "AR(Beta=`altNull3')"
104.         qui putexcel A25 = "P-Value(Beta=`altNull3')"
105.         qui putexcel A26 = "AR(Beta=`altNull4')"
106.         qui putexcel A27 = "P-Value(Beta=`altNull4')"
107.
153 .
154 .
155 .         testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `tempCovar
    > ' `primaryNull'
108.
156 .         qui putexcel B17 = `r(arStat)'
109.         qui putexcel B18 = `r(pVal)'
110.         qui putexcel B19 = `r(beta)'
111.
157 .         testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `tempCovar
    > ' `altNull'
112.         qui putexcel B20 = `r(arStat)'
113.         qui putexcel B21 = `r(pVal)'
114.
158 .         testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `tempCovar
    > ' `altNull2'
115.         qui putexcel B22 = `r(arStat)'
116.         qui putexcel B23 = `r(pVal)'
117.
159 .         if("`altNull2'"!="`altNull3'"){
118.                 qui putexcel A24 = "AR(Beta=`altNull3')"
119.                 qui putexcel A25 = "P-Value(Beta=`altNull3')"
120.
160 .                 testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `t
    > empCovar' `altNull3'
121.                 qui putexcel B24 = `r(arStat)'
122.                 qui putexcel B25 = `r(pVal)'
123.         }
124.         if("`altNull3'"!="`altNull4'"){
125.                 qui putexcel A26 = "AR(Beta=`altNull4')"
126.                 qui putexcel A27 = "P-Value(Beta=`altNull4')"
127.
```

```
161 .                 testNull `tempPi' `tempDelta' `tempPiVar' `tempDeltaVar' `t
    > empCovar' `altNull4'
    128.                 qui putexcel B26 = `r(arStat)'
    129.                 qui putexcel B27 = `r(pVal)'
    130.         }
    131.
162 .
163 . end

164 .
165 . //Construct table for LaTex output
166 . program constructTable
      1.         version 16.1
      2.
167 .         gettoken tableLabel 0:0, parse(";")
      3.         gettoken com 0:0, parse(";")
      4.         gettoken outcomeLabel 0:0, parse(";")
      5.         gettoken com 0:0, parse(";")
      6.         gettoken models 0:0, parse(";")
      7.         gettoken com 0:0, parse(";")
      8.         gettoken modelTitles 0:0, parse(";")
      9.         gettoken com 0:0, parse(";")
     10.         gettoken primaryNull 0:0, parse(";")
     11.         gettoken com 0:0, parse(";")
     12.         gettoken altNull 0:0, parse(";")
     13.         gettoken com 0:0, parse(";")
     14.         gettoken altNull2 0:0, parse(";")
     15.         gettoken com 0:0, parse(";")
     16.         gettoken altNull3 0:0, parse(";")
     17.         gettoken com 0:0, parse(";")
     18.         gettoken altNull4 0:0, parse(";")
     19.
168 .
169 .         if("`c(os)'"=="Windows"){
     20.                 local tableSaveDir = "Tables\"
     21.                 local plotSaveDir = "Plots\"
     22.                 local pythonSaveDir = "PythonScripts\"
     23.         }
     24.         else{
     25.                 local tableSaveDir = "Tables/"
     26.                 local plotSaveDir = "Plots/"
     27.                 local pythonSaveDir = "PythonScripts/"
     28.         }
     29.
```

```
170 .         local i = 1
     30.
171 .         local floatFormat = "%04.3f"
     31.
172 .         foreach model in `models'{
     32.                 di "`model'"
     33.                 preserve
     34.
173 .                 clear
     35.
174 .                 qui import excel stat=A arg1=B arg2=C arg3=D arg4=E using "
    > `tableSaveDir'iv.xlsx", sheet("`model'") cellrange(A1:E29)
     36.
175 .
176 .                 gen y = 1
     37.                 gen x = 1
     38.
177 .                 qui reg y x
     39.                 qui eststo `model'
     40.
178 .                 estadd local groupFE = "Y"
     41.                 estadd local timeFE = "Y"
     42.                 estadd local timeTrend = "Y"
     43.                 estadd local commonControls = "Y"
     44.
179 .
180 .                 //Geta Beta point estimate
181 .                 qui sum arg1 if stat=="beta"
     45.                 qui estadd local betaPoint = string(round(r(mean), .001)
    > , "`floatFormat'")
     46.                 qui estadd local modelNum = "(`i')"
     47.
182 .                 if(`i'==1){
     48.                         qui estadd local dsc = "None"
     49.                 }
     50.                 else if(`i'==2){
     51.                         qui estadd local dsc = "Documented"
     52.                 }
     53.                 else if(`i'==3){
     54.                         qui estadd local dsc = "\makecell{Large \nextlin
    > e Undocumented}"
     55.                 }
     56.                 else if(`i'==4){
     57.                         qui estadd local dsc = "\makecell{Small \nextlin
    > e Undocumented}"
     58.                 }
```

```
     59.
183 .                 local i = `i'+1
     60.
184 .                 local floatFormat = "%04.3f"
     61.
185 .                 //Get P-Value and Stars for third alternate null (if approp
    > riate)
186 .                 if("`tableLabel'"== "GasUseiv" || "`tableLabel'"=="GasUsePe
    > rCapitaiv"){
     62.                         //Get primary null hypothesis test info
187 .                         //Get AR-Stat primary null
188 .
189 .                         qui sum arg1 if stat=="P-Value(Beta=11.0)"
     63.                         qui local tempP = r(mean)
     64.                         //Get P-Value and Stars for primary null
190 .                         qui estadd local primaryP = string(round(r(mean),0.
    > 001), "`floatFormat'")
     65.                         qui sum arg1 if stat=="AR(Beta=11.0)"
     66.                         //Determine Stars for primary null
191 .                         qui local stars = ""
     67.
192 .                         if(`tempP'<0.01){
     68.                                 local stars = "***"
     69.                         }
     70.                         else if(`tempP'<0.05){
     71.                                 local stars = "**"
     72.                         }
     73.                         else if(`tempP'<0.1){
     74.                                 local stars = "*"
     75.                         }
     76.                         qui estadd local primaryARStat = string(round(r(
    > mean),.001), "`floatFormat'")+"`stars'"
     77.                         qui sum arg1 if stat=="P-Value(Beta=9.7)"
     78.                         qui local tempP = r(mean)
     79.                         //Get P-Value and Stars for primary null
```

```
193 .                         qui estadd local altP = string(round(r(mean),0.001)
    > , "`floatFormat'")
     80.
194 .                         qui sum arg1 if stat=="AR(Beta=9.7)"
     81.                            //Determine Stars for primary null
195 .                         qui local stars = ""
     82.
196 .                         if(`tempP'<0.01){
     83.                                    local stars = "***"
     84.                            }
     85.                            else if(`tempP'<0.05){
     86.                                    local stars = "**"
     87.                            }
     88.                            else if(`tempP'<0.1){
     89.                                    local stars = "*"
     90.                            }
     91.                            qui estadd local altARStat = string(round(r(mean
    > ),.001), "`floatFormat'")+"`stars'"
     92.                            //Get alternate null hypothesis test info
197 .                         //Get AR-Stat alternate null
198 .                         qui sum arg1 if stat=="P-Value(Beta=1)"
     93.                            qui local tempP = r(mean)
     94.                            //Get P-Value and Stars for alternate null
199 .                         qui estadd local altP2 = string(round(r(mean),0.001
    > ), "`floatFormat'")
     95.                            qui sum arg1 if stat=="AR(Beta=1)"
     96.                            //Determine Stars for alternate null
200 .                         qui local stars = ""
     97.
201 .                         if(`tempP'<0.01){
     98.                                    local stars = "***"
     99.                            }
    100.                            else if(`tempP'<0.05){
    101.                                    local stars = "**"
    102.                            }
    103.                            else if(`tempP'<0.1){
    104.                                    local stars = "*"
    105.                            }
    106.                            qui estadd local altARStat2 = string(round(r(mea
    > n),.001), "`floatFormat'")+"`stars'"
    107.
```

```
202 .                         //Get P-Value and Stars for second alternate null
203 .                         qui sum arg1 if stat=="P-Value(Beta=-4.8)"
    108.                         qui local tempP = r(mean)
    109.                         qui estadd local altP3 = string(round(r(mean),0.
> 001), "`floatFormat'")
    110.                         qui sum arg1 if stat=="AR(Beta=-4.8)"
    111.                         //Determine Stars for alternate null
204 .                         qui local stars = ""
    112.
205 .                         if(`tempP'<0.01){
    113.                                 local stars = "***"
    114.                         }
    115.                         else if(`tempP'<0.05){
    116.                                 local stars = "**"
    117.                         }
    118.                         else if(`tempP'<0.1){
    119.                                 local stars = "*"
    120.                         }
    121.                         qui estadd local altARStat3 = string(round(r(mea
> n),.001), "`floatFormat'")+"`stars'"
    122.
206 .                         //Get P-Value and Stars for second alternate null
207 .                         qui sum arg1 if stat=="P-Value(Beta=-6.1)"
    123.                         qui local tempP = r(mean)
    124.                         qui estadd local altP4 = string(round(r(mean),0.
> 001), "`floatFormat'")
    125.                         qui sum arg1 if stat=="AR(Beta=-6.1)"
    126.                         //Determine Stars for alternate null
208 .                         qui local stars = ""
    127.
209 .                         if(`tempP'<0.01){
    128.                                 local stars = "***"
    129.                         }
    130.                         else if(`tempP'<0.05){
    131.                                 local stars = "**"
    132.                         }
    133.                         else if(`tempP'<0.1){
    134.                                 local stars = "*"
    135.                         }
    136.                         qui estadd local altARStat4 = string(round(r(mea
> n),.001), "`floatFormat'")+"`stars'"
    137.                 }
    138.                 else{
    139.
```

```
210 .                         //Get primary null hypothesis test info
211 .                         //Get AR-Stat primary null
212 .
213 .                         qui sum arg1 if stat=="P-Value(Beta=1.6)"
    140.                         qui local tempP = r(mean)
    141.                         //Get P-Value and Stars for primary null
214 .                         qui estadd local primaryP = string(round(r(mean),0.
    > 001), "`floatFormat'")
    142.                         qui sum arg1 if stat=="AR(Beta=1.6)"
    143.                         //Determine Stars for primary null
215 .                         qui local stars = ""
    144.
216 .                         if(`tempP'<0.01){
    145.                                 local stars = "***"
    146.                         }
    147.                         else if(`tempP'<0.05){
    148.                                 local stars = "**"
    149.                         }
    150.                         else if(`tempP'<0.1){
    151.                                 local stars = "*"
    152.                         }
    153.                         qui estadd local primaryARStat = string(round(r(
    > mean),.001), "`floatFormat'")+"`stars'"
    154.                         //Get alternate null hypothesis test info
217 .                         //Get AR-Stat alternate null
218 .                         qui sum arg1 if stat=="P-Value(Beta=1.0)"
    155.                         qui local tempP = r(mean)
    156.                         //Get P-Value and Stars for alternate null
219 .                         qui estadd local altP = string(round(r(mean),0.001)
    > , "`floatFormat'")
    157.                         qui sum arg1 if stat=="AR(Beta=1.0)"
    158.                         //Determine Stars for alternate null
220 .                         qui local stars = ""
    159.
```

```
221 .                         if(`tempP'<0.01){
    160.                                 local stars = "***"
    161.                            }
    162.                            else if(`tempP'<0.05){
    163.                                 local stars = "**"
    164.                            }
    165.                            else if(`tempP'<0.1){
    166.                                 local stars = "*"
    167.                            }
    168.                            qui estadd local altARStat = string(round(r(mean
    > ),.001), "`floatFormat'")+"`stars'"
    169.
222 .                         //Get alternate null hypothesis test info
223 .                         //Get AR-Stat alternate null
224 .                         qui sum arg1 if stat=="P-Value(Beta=.3)"
    170.                            qui local tempP = r(mean)
    171.                            //Get P-Value and Stars for alternate null
225 .                         qui estadd local altP2 = string(round(r(mean),0.001
    > ), "`floatFormat'")
    172.                            qui sum arg1 if stat=="AR(Beta=.3)"
    173.                            //Determine Stars for alternate null
226 .                         qui local stars = ""
    174.
227 .                         if(`tempP'<0.01){
    175.                                 local stars = "***"
    176.                            }
    177.                            else if(`tempP'<0.05){
    178.                                 local stars = "**"
    179.                            }
    180.                            else if(`tempP'<0.1){
    181.                                 local stars = "*"
    182.                            }
    183.                            qui estadd local altARStat2 = string(round(r(mea
    > n),.001), "`floatFormat'")+"`stars'"
    184.
```

```
228 .                         //Get alternate null hypothesis test info
229 .                         //Get AR-Stat alternate null
230 .                         qui sum arg1 if stat=="P-Value(Beta=0)"
    185.                            qui local tempP = r(mean)
    186.                            //Get P-Value and Stars for alternate null
231 .                         qui estadd local altP3 = string(round(r(mean),0.001
    > ), "`floatFormat'")
    187.                            qui sum arg1 if stat=="AR(Beta=0)"
    188.                            //Determine Stars for alternate null
232 .                         qui local stars = ""
    189.
233 .                         if(`tempP'<0.01){
    190.                                    local stars = "***"
    191.                            }
    192.                            else if(`tempP'<0.05){
    193.                                    local stars = "**"
    194.                            }
    195.                            else if(`tempP'<0.1){
    196.                                    local stars = "*"
    197.                            }
    198.                            qui estadd local altARStat3 = string(round(r(mea
    > n),.001), "`floatFormat'")+"`stars'"
    199.
234 .                         //Get P-Value and Stars for second alternate null
235 .                         qui sum arg1 if stat=="P-Value(Beta=-2.5)"
    200.                            qui local tempP = r(mean)
    201.                            qui estadd local altP4 = string(round(r(mean),0.
    > 001), "`floatFormat'")
    202.                            qui sum arg1 if stat=="AR(Beta=-2.5)"
    203.                            //Determine Stars for alternate null
236 .                         qui local stars = ""
    204.
237 .                         if(`tempP'<0.01){
    205.                                    local stars = "***"
    206.                            }
    207.                            else if(`tempP'<0.05){
    208.                                    local stars = "**"
    209.                            }
    210.                            else if(`tempP'<0.1){
    211.                                    local stars = "*"
    212.                            }
    213.                            qui estadd local altARStat4 = string(round(r(mea
    > n),.001), "`floatFormat'")+"`stars'"
    214.                    }
    215.                    //Get N
```

```
238 .                 qui sum arg1 if stat=="N"
    216.                 qui estadd local obs = string(round(r(mean)), "%9.0fc")
    217.
239 .                 //Get 95% Confidence set
240 .                 local floatFormat = "%3.2f"
    218.                 qui sum arg3 if stat=="95% Confidence Set"
    219.                 if(r(N)==1){
    220.                         qui sum arg2 if stat=="95% Confidence Set"
    221.                         local lPoint = string(round(r(mean),.01), "`floa
    > tFormat'")
    222.                         sum arg3 if stat=="95% Confidence Set"
    223.                         qui local rPoint = string(round(r(mean),.01), "`
    > floatFormat'")
    224.
241 .                         qui estadd local cs95 "\makecell{(-$\infty$, $`lPoi
    > nt']$; \nextline $[`rPoint'$, $\infty$)}"
    225.
242 .                 }
    226.                 else{
    227.                         qui sum arg1 if stat=="95% Confidence Set"
    228.                         if(r(N)==1){
    229.                                 local lPoint = string(round(r(mean),.01)
    > , "`floatFormat'")
    230.                         }
    231.                         else{
    232.                                 local lPoint = "-$\infty$"
    233.                         }
    234.                         qui sum arg2 if stat=="95% Confidence Set"
    235.                         if(r(N)==1){
    236.                                 local rPoint = string(round(r(mean),.01)
    > , "`floatFormat'")
    237.                         }
    238.                         else{
    239.                                 local rPoint = "$\infty$"
    240.                         }
    241.
```

```
243 .                         if("`lPoint'"=="-$\infty$" & "`rPoint'"=="$\infty$"
    > ){
    242.                                qui estadd local cs95 = "(`lPoint', `rPo
    > int')"
    243.                         }
    244.                         else{
    245.                                qui estadd local cs95 = "[`lPoint', `rPo
    > int']"
    246.                         }
    247.
244 .
245 .
246 .                 }
    248.
247 .                 //Get 90% Confidence Set
248 .                 qui sum arg3 if stat=="90% Confidence Set"
    249.                    if(r(N)==1){
    250.                           qui sum arg2 if stat=="90% Confidence Set"
    251.                           local lPoint = string(round(r(mean),.01))
    252.                           qui sum arg3 if stat=="90% Confidence Set"
    253.                           local rPoint = string(round(r(mean),.01))
    254.
249 .                         qui estadd local cs90 "(-$\infty$, `lPoint']; [`rPo
    > int', $\infty$)"
    255.
250 .                 }
    256.                    else{
    257.                           qui sum arg1 if stat=="90% Confidence Set"
    258.                           if(r(N)==1){
    259.                                  local lPoint = string(round(r(mean),.01)
    > )
    260.                           }
    261.                           else{
    262.                                  local lPoint = "-$\infty$"
    263.                           }
    264.                           qui sum arg2 if stat=="90% Confidence Set"
    265.                           if(r(N)==1){
    266.                                  local rPoint = string(round(r(mean),.01)
    > )
    267.                           }
    268.                           else{
    269.                                  local rPoint = "$\infty$"
    270.                           }
    271.
```

```
251 .                         qui estadd local cs90 = "[`lPoint', `rPoint']"
    272.
252 .                 }
    273.
253 .
254 .
255 .
256 .                 restore
    274.         }
    275.
257 .         //Output Table
258 .
259 .         if("`c(os)'"=="Windows"){
    276.                 local tableSaveDir = "Tables\IV\"
    277.                 local plotSaveDir = "Plots\"
    278.                 local pythonSaveDir = "PythonScripts\"
    279.         }
    280.         else{
    281.                 local tableSaveDir = "Tables/IV/"
    282.                 local plotSaveDir = "Plots/"
    283.                 local pythonSaveDir = "PythonScripts/"
    284.         }
    285.
260 .
261 .         if("`tableLabel'"=="VMTiv"){
    286.                 local titleOutcome = "Log Registrations on Log VMT"
    287.                 local noteString = "\hline \hline \end{tabular} \vspace{
    > 0.2cm} \begin{tablenotes} \begin{spacing}{0} \begin{footnotesize} \item \te
    > xtbf{Notes:} The top panel of this table presents IV estimates and 95\% con
    > fidence sets robust to weak instruments of the linear relationship between
    > log vehicle-miles travelled (VMT) and log vehicle registrations.  The botto
    > m panel presents Anderson-Rubin chi-squared statistics along with p-values
    > for different null hypotheses pertaining to the parameter estimated in the
    > top panel.  The unit of observation is state-year.  All regressions include
    >  state fixed effects, year fixed effects and state-specific linear time tre
    > nds. The four specifications pertaining to different levels of ``Data Sourc
    > e Controls'' (DSCs) and control specifications are documented in Table \cre
    > f{ssec:Spec}. Confidence sets are based on standard errors clustered by sta
    > te. * p\textless 0.1, ** p\textless 0.05, *** p\textless 0.01 \end{footnote
    > size} \end{spacing} \end{tablenotes} \end{table}"
    288.         }
    289.         else if("`tableLabel'"=="VMTPerCapitaiv"){
    290.                 local titleOutcome = "Log Registrations per Capita on Lo
    > g VMT per Capita"
    291.                 local noteString = "\hline \hline \end{tabular} \vspace{
    > 0.2cm} \begin{tablenotes} \begin{spacing}{0} \begin{footnotesize} \item \te
```

```
> xtbf{Notes:} The top panel of this table presents IV estimates and 95\% con
> fidence sets robust to weak instruments of the linear relationship between
> log vehicle-miles travelled (VMT) per capita and log vehicle registrations
> per capita.  The bottom panel presents Anderson-Rubin chi-squared statistic
> s along with p-values for different null hypotheses pertaining to the param
> eter estimated in the top panel.  The unit of observation is state-year. Th
> e four specifications pertaining to different levels of ``Data Source Contr
> ols'' (DSCs) and control specifications are documented in Table \cref{ssec:
> Spec}. Confidence sets are based on standard errors clustered by state. * p
> \textless 0.1, ** p\textless 0.05, *** p\textless 0.01 \end{footnotesize} \
> end{spacing} \end{tablenotes} \end{table}"
292.         }
293.         else if("`tableLabel'"=="GasUseiv"){
294.                 local titleOutcome = "Log Registrations on Log Highway G
> as Use"
295.                 local noteString = "\hline \hline \end{tabular} \vspace{
> 0.2cm} \begin{tablenotes} \begin{spacing}{0} \begin{footnotesize} \item \te
> xtbf{Notes:} The top panel of this table presents IV estimates and 95\% con
> fidence sets robust to weak instruments of the linear relationship between
> log highway gas use and log vehicle registrations.  The bottom panel presen
> ts Anderson-Rubin chi-squared statistics along with p-values for different
> null hypotheses pertaining to the parameter estimated in the top panel. The
>  unit of observation is state-year. The four specifications pertaining to d
> ifferent levels of ``Data Source Controls'' (DSCs) and control specificatio
> ns are documented in Table \cref{ssec:Spec}. Confidence sets are based on s
> tandard errors clustered by state. * p\textless 0.1, ** p\textless 0.05, **
> * p\textless 0.01 \end{footnotesize} \end{spacing} \end{tablenotes} \end{ta
> ble}"
296.         }
297.         else if("`tableLabel'"=="GasUsePerCapitaiv"){
298.                 local titleOutcome = "Log Registrations per Capita on Lo
> g Highway Gas Use per Capita"
299.                 local noteString = "\hline \hline \end{tabular} \vspace{
> 0.2cm} \begin{tablenotes} \begin{spacing}{0} \begin{footnotesize} \item \te
> xtbf{Notes:} The top panel of this table presents IV estimates and 95\% con
> fidence sets robust to weak instruments of the linear relationship between
> log highway gas use per capita and log vehicle registrations per capita.  T
> he bottom panel presents Anderson-Rubin chi-squared statistics along with p
> -values for different null hypotheses pertaining to the parameter estimated
>  in the top panel.  The unit of observation is state-year.  The four specif
> ications pertaining to different levels of ``Data Source Controls'' (DSCs)
> and control specifications are documented in Table \cref{ssec:Spec}. Confid
> ence sets are based on standard errors clustered by state. * p\textless 0.1
> , ** p\textless 0.05, *** p\textless 0.01 \end{footnotesize} \end{spacing}
> \end{tablenotes} \end{table}"
300.         }
```

```
    301.
262 .         local saveFile = "`tableSaveDir'"+"`tableLabel'"+".tex"
    302.
263 .         local fmtDiD = `"drop(*) nocon label compress starlevels(* 0.1 ** 0
    > .05 *** 0.01) nodepvar nomtitle nonumbers not nostar noline"'
    303.
264 .
265 .
266 .
267 .         if("`tableLabel'"== "GasUseiv" || "`tableLabel'"=="GasUsePerCapitai
    > v"){
    304.                 local greekLetter = "\gamma"
    305.
268 .
269 .
270 .                 esttab `models' using "`saveFile'", drop(*) stats(panel1 mo
    > delNum betaPoint cs95 dsc groupFE timeFE timeTrend commonControls obs panel
    > 2 primaryARStat primaryP space1 altARStat altP space2 altARStat2 altP2 spac
    > e3 altARStat3 altP3 space4 altARStat4 altP4, labels( "\hline \hline \multic
    > olumn{5}{l}{\textit{Panel A: IV Estimates and Robust 95\% Confidence Interv
    > als}} \\" "\multicolumn{5}{c}{Dep. Var.: Log Number of Vehicle Registration
    > s} \\ " "$`greekLetter'$" "95\% Conf. Set" "\makecell[l]{Data Source \nextl
    > ine Controls}" "State Fixed Effects" "Year Fixed Effects" "\makecell[l]{Sta
    > te-Specific Linear \\ Time Trends}" "Common Controls" "Number of Obs." "\hl
    > ine \hline \multicolumn{5}{l}{\textit{Panel B: Anderson-Robin Test Statisti
    > cs and P-Values}} \\" " \multicolumn{5}{c}{\underline{\$ H_{0}:`greekLetter
    > '=11\$}} \\ AR Stat" "p-value" " " " \multicolumn{5}{c}{\underline{\$ H_{0}
    > :`greekLetter'=9.7\$}} \\ AR Stat" "p-value" " " "\multicolumn{5}{c}{\under
    > line{\$ H_{0}:`greekLetter'=\$1}} \\ AR Stat" "p-value" " " "\multicolumn{5
    > }{c}{\underline{\$ H_{0}:`greekLetter'=\$-4.8}} \\ AR Stat" "p-value" " " "
    > \multicolumn{5}{c}{\underline{\$ H_{0}:`greekLetter'=\$-6.1}} \\ AR Stat" "
    > p-value")) nostar nomtitle nonumbers not nolines title("Robust IV Estimates
    >  of `titleOutcome' \label{tab:`tableLabel'}") replace postfoot("`noteString
    > '") sub(\_ _)
    306.
```

```
271 .         }
    307.         else{
    308.                 local greekLetter = "\beta"
    309.
272 .                 esttab `models' using "`saveFile'", drop(*) stats(panel1 mo
    > delNum betaPoint cs95 dsc groupFE timeFE timeTrend commonControls obs panel
    > 2 primaryARStat primaryP space1 altARStat altP space2 altARStat2 altP2 spac
    > e3 altARStat3 altP3 space4 altARStat4 altP4, labels("\hline \hline \multico
    > lumn{5}{l}{\textit{Panel A: IV Estimates and Robust 95\% Confidence Interva
    > ls}} \\" "\multicolumn{5}{c}{Dep. Var.: Log Number of Vehicle Registrations
    > } \\ " "$`greekLetter'$" "95\% Conf. Set" "\makecell[l]{Data Source \nextli
    > ne Controls}" "State Fixed Effects" "Year Fixed Effects" "\makecell[l]{Stat
    > e-Specific Linear \\ Time Trends}" "Common Controls" "Number of Obs." "\hli
    > ne \hline \multicolumn{5}{l}{\textit{Panel B: Anderson-Robin Test Statistic
    > s and P-Values}} \\" "\multicolumn{5}{c}{\underline{\$ H_{0}:`greekLetter'=
    > 1.6\$}} \\ AR Stat" "p-value" " " "\multicolumn{5}{c}{\underline{\$ H_{0}:`
    > greekLetter'=1\$}} \\ AR Stat" "p-value" " " "\multicolumn{5}{c}{\underline
    > {\$ H_{0}:`greekLetter'=0.3\$}} \\ AR Stat" "p-value" " " "\multicolumn{5}{
    > c}{\underline{\$ H_{0}:`greekLetter'=0\$}} \\ AR Stat" "p-value" " " "\mult
    > icolumn{5}{c}{\underline{\$ H_{0}:`greekLetter'=-2.5\$}} \\ AR Stat" "p-val
    > ue") fmt(%4.0fc)) nostar nomtitle nonumbers not nolines title("Robust IV Es
    > timates of Log Registration on `titleOutcome' \label{tab:`tableLabel'}") re
    > place postfoot("`noteString'") sub(\_ _)
    310.
273 .
274 .         }
    311.
275 .
276 .
277 .
278 . end

279 .
    end of do-file
```

```
280 .
281 . // // //Execute Stata Do File to load and clean the data for VMT models
282 . do LoadDataVMT

283 . clear

284 . //Load Data for travel analysis
285 .
286 .
287 .
288 .
289 . if("`c(os)'"=="Windows"){
290 .         local dataFile = "CleanedData\FHWACleanV4.csv"
291 . }

292 . else{
293 .         local dataFile = "CleanedData/FHWACleanV4.csv"
294 . }

295 .
296 . import delimited using "`dataFile'"
    (encoding automatically selected: ISO-8859-1)
    (90 vars, 3,009 obs)

297 .
298 . //Add State Codes to all states
299 . sort state

300 . egen tempStateCode = mode(statecode), by(state)

301 . replace statecode = tempStateCode
    (357 real changes made)

302 . drop tempStateCode
```

```
303 .
304 . //Label Variables
305 . label var logregistrations "Log Total Registrations"

306 . label var logvmt "Log VMT"

307 . label var loglicenseddrivers "Log Licensed Drivers"

308 . label var logjointpopulation "Log Total Population"

309 . label var logrealmeangasprice "Log Mean Gas Price"

310 . label var logemployment  "Log Employment"

311 . label var logrealjointincome  "Log Total Income"

312 . label var logmetropop  "Log (1 + Metro Population)"

313 . label var lognonmetropop  "Log (1 + Non-Metro Population)"

314 . label var logmetrorealinc  "Log (1 + Metro Income)"

315 . label var lognonmetrorealinc  "Log (1 + Non-Metro Income)"

316 . label var logrealstategdp  "Log State GDP"

317 . label var logpopulation  "Log Population"

318 . label var logrealtotalincome  "Log Total Income"

319 . label var logroadmileage "Log Road Mileage"

320 . label var nosafetyind "Treatment"
```

```
321 .
322 .
323 . //Create Inverse Hyperbolic Sine Variables
324 . gen asinhMetroInc = asinh(realmetroincome2018m)
    (459 missing values generated)

325 . gen asinhNonMetroInc = asinh(realnonmetroincome2018m)
    (459 missing values generated)

326 .
327 . gen asinhMetroPop = asinh(metropopulation)
    (459 missing values generated)

328 . gen asinhNonMetroPop = asinh(nonmetropopulation)
    (459 missing values generated)

329 .
330 . //Per capita model
331 . //construct outcome variable
332 . gen logRegPerCapita = logregistrations-log(metropopulation + nonmetropopula
    > tion)
    (510 missing values generated)

333 . gen logVMTPerCapita = logvmt-log(metropopulation + nonmetropopulation)
    (510 missing values generated)

334 .
335 . //Label Asinh variables
336 . label var asinhMetroInc "Asinh(Metro Income)"

337 . label var asinhNonMetroInc "Asinh(Non-Metro Income)"

338 .
339 . label var asinhMetroPop "Asinh(Metro Population)"
```

```
340 . label var asinhNonMetroPop "Asinh(Non-Metro Population)"

341 .
342 . //Label treatment variables
343 .
344 . label define nosafetyind 0 "No Treatment" 1 "Treatment"

345 . label values nosafetyind nosafetyind

346 .
347 .
348 . //Generate State Dummies
349 . egen stateGroup = group(state)

350 .
351 . //Drop Datapoints that are replications of past data for registrations
352 . drop if statecode=="CO" & year==2006
    (1 observation deleted)

353 . drop if statecode=="IN" & (year==2006 | year==2007 | year==2009)
    (3 observations deleted)

354 . drop if statecode=="MT" & year==2005
    (1 observation deleted)

355 . drop if statecode=="NJ" & year==2008
    (1 observation deleted)

356 . drop if statecode=="PR" & (year==2002 | year==2003 | year==2005 | year==200
    > 8 | year==2009 | year==2010)
    (0 observations deleted)

357 . drop if statecode=="TX" & year==2009
    (1 observation deleted)
```

```
358 . drop if statecode=="IL" & year==2011
    (1 observation deleted)

359 . drop if statecode=="NH" & year==2012
    (1 observation deleted)

360 . drop if statecode=="NY" & year==2012
    (1 observation deleted)

361 . //Drop Datapoints that are replications of past data for VMT
362 . drop if statecode=="MO" & year == 2003
    (1 observation deleted)

363 . drop if statecode=="IN" & (year == 2004 | year==2009)
    (1 observation deleted)

364 . drop if statecode=="NV" & year == 2004
    (1 observation deleted)

365 . drop if statecode=="NH" & year == 2004
    (1 observation deleted)

366 . drop if statecode=="NY" & year == 2005
    (1 observation deleted)

367 . drop if statecode=="AZ" & year == 2009
    (1 observation deleted)

368 . drop if statecode=="WY" & year == 2010
    (1 observation deleted)

369 . drop if statecode=="PR" & (year == 2011 | year == 2012 | year == 2013 | yea
    > r == 2015 | year == 2016)
    (0 observations deleted)
```

```
370 .
371 . //Drop if data is prior to 1970
372 . drop if year<1970
    (510 observations deleted)

373 .
374 . //Normalize differenece of outcome variables for identification of extreme
    > points
375 .
376 . sort stateGroup year

377 . xtset stateGroup year

    Panel variable: stateGroup (unbalanced)
     Time variable: year, 1970 to 2018, but with gaps
             Delta: 1 unit

378 .
379 . gen DiffLogReg = d.logregistrations
    (118 missing values generated)

380 . gen DiffLogVMT = d.logvmt
    (118 missing values generated)

381 . gen DiffLogRegPerCapita = d.logRegPerCapita
    (118 missing values generated)

382 . gen DiffLogVMTPerCapita = d.logVMTPerCapita
    (118 missing values generated)

383 .
384 . local normVarList = "DiffLogReg DiffLogVMT DiffLogRegPerCapita DiffLogVMTPe
    > rCapita"
```

```
385 .
386 . sort stateGroup

387 .
388 . foreach varToNorm in `normVarList'{
      2.
389 .         by stateGroup: egen tempMean = mean(`varToNorm')
      3.         by stateGroup: egen tempSD = sd(`varToNorm')
      4.         by stateGroup: egen tempNumData = count(`varToNorm')
      5.         gen tempSE = tempSD/sqrt(tempNumData)
      6.
390 .         gen norm`varToNorm' = (`varToNorm'-tempMean)/tempSD
      7.
391 .         drop tempMean tempSD tempNumData tempSE
      8.
392 .         di "`varToNorm'"
      9.
393 . }
    (118 missing values generated)
    DiffLogReg
    (118 missing values generated)
    DiffLogVMT
    (118 missing values generated)
    DiffLogRegPerCapita
    (118 missing values generated)
    DiffLogVMTPerCapita

394 .
395 . //Generate Colorado Dummy
396 . gen coDum1 = 0

397 . gen coDum2 = 0

398 . replace coDum1 = 1 if statecode=="CO" & year>=2002 & year<=2009
    (7 real changes made)
```

```
399 . replace coDum2 = 1 if statecode=="CO" & year>=2010
    (9 real changes made)

400 .
401 . drop if coDum1 == 1
    (7 observations deleted)

402 .
403 .
404 . //Generate dummy variable for data footnotes
405 . gen transactionDataDummy = 0

406 .
407 . replace transactionDataDummy = 1 if statecode=="AR" & year>=2011
    (8 real changes made)

408 . replace transactionDataDummy = 2 if statecode=="GA" & year>=2011
    (8 real changes made)

409 . replace transactionDataDummy = 3 if statecode=="IA" & year>=2011
    (8 real changes made)

410 . replace transactionDataDummy = 4 if statecode=="IL" & year>=2012
    (7 real changes made)

411 . replace transactionDataDummy = 5 if statecode=="KY" & year>=2011
    (8 real changes made)

412 . replace transactionDataDummy = 6 if statecode=="LA" & year>=2012
    (7 real changes made)

413 . replace transactionDataDummy = 7 if statecode=="ME" & year>=2011 //Maine is
    >  a state with what looks to not have a consistent change
    (8 real changes made)
```

```
414 . replace transactionDataDummy = 8 if statecode=="ME" & year>2014
    (4 real changes made)

415 . replace transactionDataDummy = 9 if statecode=="MI" & year>=2011 //Michigan
    >  is a state with what looks to not have a consistent change
    (8 real changes made)

416 . replace transactionDataDummy = 10 if statecode=="MI" & year>2012
    (6 real changes made)

417 . replace transactionDataDummy = 11 if statecode=="MN" & year>=2011
    (8 real changes made)

418 . replace transactionDataDummy = 12 if statecode=="NV" & year>=2011
    (8 real changes made)

419 . replace transactionDataDummy = 13 if statecode=="OK" & year>=2011
    (8 real changes made)

420 . replace transactionDataDummy = 14 if statecode=="SD" & year>=2011
    (8 real changes made)

421 . replace transactionDataDummy = 15 if statecode=="TN" & year>=2011
    (8 real changes made)

422 . replace transactionDataDummy = 16 if statecode=="TX" & year>=2011
    (8 real changes made)

423 . replace transactionDataDummy = 17 if statecode=="WA" & year>=2011
    (8 real changes made)

424 . replace transactionDataDummy = 18 if statecode=="WI" & year>=2011
    (8 real changes made)
```

```
425 . replace transactionDataDummy = 19 if statecode=="WY" & year>=2011
    (8 real changes made)

426 .
427 .
428 .
429 .
430 . //Local Extreme Discontinuity controls
431 .
432 . sort state year

433 .
434 . python
    ──────────────────────────────────────────────── python (type end to exit) ───
    >>>
    >>> import numpy as np
    >>> import pandas as p
    >>> from sfi import Data
    >>>
    >>>
    >>> def getDSCDummies(data, col, cutoff):
    ...
    ...         tempExtState = None
    ...
    ...         tempIndex = 0
    ...         stateCol = 0
    ...
    ...         rows, cols = data.shape
    ...
    ...         dummyExt = []
    ...
    ...         for i in range(rows):
    ...                 tempDiff = float(data[i,col])
    ...                 tempState = data[i,stateCol]
    ...
    ...
    ...                 if(abs(tempDiff)>cutoff):
    ...                         tempExtState = tempState
    ...                         tempIndex+=1
    ...
    ...                 if tempState==tempExtState:
    ...                         dummyExt.append(tempIndex)
    ...                 else:
    ...                         dummyExt.append(0)
    ...
    ...
```

```
...         dummyExt = np.array(dummyExt).reshape((-1,1))
...         return dummyExt
...
>>> dataCols = ['state', 'year', 'normDiffLogReg', 'normDiffLogVMT', 'normDif
> fLogRegPerCapita', 'normDiffLogVMTPerCapita']
>>>
>>> data = np.array(Data.get(dataCols, missingval=np.nan))
>>>
>>> print(data)
[['Alabama' '1970' 'nan' 'nan' 'nan' 'nan']
 ['Alabama' '1971' '1.0640671570008131' '0.3361620217656708'
  '0.9150116516585626' '0.16061868662892634']
 ['Alabama' '1972' '1.0231103264075507' '0.24914037544421433'
  '0.9109946118213749' '0.11750494756107976']
 ...
 ['Wyoming' '2016' '0.45763583139776387' '-1.6128834808565655'
  '0.7516965179523265' '-1.1845977355288309']
 ['Wyoming' '2017' '-1.1510098900675998' '0.6274459465980643'
  '-0.8234171123760192' '1.2249676800199891']
 ['Wyoming' '2018' 'nan' 'nan' 'nan' 'nan']]
>>>
>>> regCutoffLow = 2.93 #Spec. 4
>>> regCutoffPrimary = 4.02 #Spec. 3
>>> regCutoffHigh = 7 #Spec. 2
>>>
>>> vmtCutoffLow = 2.93 #Spec. 4
>>> vmtCutoffPrimary = 4.02 #Spec. 3
>>> vmtCutoffHigh = 7 #Spec. 2
>>>
>>> regCol = dataCols.index('normDiffLogReg')
>>> vmtCol = dataCols.index('normDiffLogVMT')
>>>
>>> regPerCapitaCol = dataCols.index('normDiffLogRegPerCapita')
>>> vmtPerCapitaCol = dataCols.index('normDiffLogVMTPerCapita')
>>>
>>> rows, cols = data.shape
>>>
>>> regDummyExtLow = getDSCDummies(data, regCol, regCutoffLow)
>>> regDummyExtPrimary = getDSCDummies(data, regCol, regCutoffPrimary)
>>> regDummyExtHigh = getDSCDummies(data, regCol, regCutoffHigh)
>>>
>>> vmtDummyExtLow = getDSCDummies(data, vmtCol, vmtCutoffLow)
>>> vmtDummyExtPrimary = getDSCDummies(data, vmtCol, vmtCutoffPrimary)
>>> vmtDummyExtHigh = getDSCDummies(data, vmtCol, vmtCutoffHigh)
>>>
>>> regPerCapitaDummyExtLow = getDSCDummies(data, regPerCapitaCol, regCutoffL
```

```
> ow)
>>> regPerCapitaDummyExtPrimary = getDSCDummies(data, regPerCapitaCol, regCut
> offPrimary)
>>> regPerCapitaDummyExtHigh = getDSCDummies(data, regPerCapitaCol, regCutoff
> High)
>>>
>>> vmtPerCapitaDummyExtLow = getDSCDummies(data, vmtPerCapitaCol, vmtCutoffL
> ow)
>>> vmtPerCapitaDummyExtPrimary = getDSCDummies(data, vmtPerCapitaCol, vmtCut
> offPrimary)
>>> vmtPerCapitaDummyExtHigh = getDSCDummies(data, vmtPerCapitaCol, vmtCutoff
> High)
>>>
>>> Data.addVarInt('regDummyExtSpec4')
>>> Data.store("regDummyExtSpec4",None, regDummyExtLow[:])
>>> Data.addVarInt('regDummyExtSpec3')
>>> Data.store("regDummyExtSpec3",None, regDummyExtPrimary[:])
>>> Data.addVarInt('regDummyExtSpec2')
>>> Data.store("regDummyExtSpec2",None, regDummyExtHigh[:])
>>>
>>> Data.addVarInt('vmtDummyExtSpec4')
>>> Data.store("vmtDummyExtSpec4",None, vmtDummyExtLow[:])
>>> Data.addVarInt('vmtDummyExtSpec3')
>>> Data.store("vmtDummyExtSpec3",None, vmtDummyExtPrimary[:])
>>> Data.addVarInt('vmtDummyExtSpec2')
>>> Data.store("vmtDummyExtSpec2",None, vmtDummyExtHigh[:])
>>>
>>>
>>> Data.addVarInt('regPerCapitaDummyExtSpec4')
>>> Data.store("regPerCapitaDummyExtSpec4",None, regPerCapitaDummyExtLow[:])
>>> Data.addVarInt('regPerCapitaDummyExtSpec3')
>>> Data.store("regPerCapitaDummyExtSpec3",None, regPerCapitaDummyExtPrimary[
> :])
>>> Data.addVarInt('regPerCapitaDummyExtSpec2')
>>> Data.store("regPerCapitaDummyExtSpec2",None, regPerCapitaDummyExtHigh[:])
>>>
>>> Data.addVarInt('vmtPerCapitaDummyExtSpec4')
>>> Data.store("vmtPerCapitaDummyExtSpec4",None, vmtPerCapitaDummyExtLow[:])
>>> Data.addVarInt('vmtPerCapitaDummyExtSpec3')
>>> Data.store("vmtPerCapitaDummyExtSpec3",None, vmtPerCapitaDummyExtPrimary[
> :])
>>> Data.addVarInt('vmtPerCapitaDummyExtSpec2')
>>> Data.store("vmtPerCapitaDummyExtSpec2",None, vmtPerCapitaDummyExtHigh[:])
>>>
>>> end
```

```
435 .
436 .
437 . //Generate normalized year values to reduce differences in magnitude of cov
    > ariates.
438 . //Scale year for better numerical performance
439 . local yearScale = 1

440 . gen normYear = year/`yearScale'

441 . gen quadYear = (year/1)^2

442 .
443 . //Establish treatment interacted with pre-treatment population share in met
    > ro areas
444 . frame copy default MetroShares

445 . frame change MetroShares

446 . keep if year==1970
    (2,424 observations deleted)

447 .
448 . capture gen preTreatMetroShare = metropopulation/(metropopulation+nonmetrop
    > opulation)

449 .
450 . frame change default

451 .
452 . cap frlink m:1 state, frame(MetroShares)

453 . cap frget preTreatMetroShare, from(MetroShares)
```

```
454 .
455 . cap gen metroIntTreatment = nosafetyind*preTreatMetroShare

456 .
457 . frame drop MetroShares

458 .
    end of do-file

459 .
460 . // Urban-Rural models
461 . capture frame drop ruralUrban

462 . capture drop ruralUrban ruralvmt urbanvmt logruralvmt logurbanvmt

463 . do VMTIVUrbanRural

464 . eststo clear

465 .
466 .
467 . if("`c(os)'"=="Windows"){
468 .         local tableSaveDir = "ReviewResponseResults\"
469 .         local plotSaveDir = "ReviewResponseResults\"
470 .         local pythonSaveDir = "PythonScripts\"
471 . }

472 . else{
473 .         local tableSaveDir = "ReviewResponseResults/"
474 .         local plotSaveDir = "ReviewResponseResults/"
475 .         local pythonSaveDir = "PythonScripts/"
476 . }

477 .
```

```
. //Establish Control Variable Sets
. //Spec1 controls are the same for both registrations and VMT
. local spec1Controls = "c.normYear#i.stateGroup c.quadYear#i.stateGroup asin
> hMetroPop asinhNonMetroPop asinhMetroInc asinhNonMetroInc logrealmeangaspri
> ce logemployment loglicenseddrivers logrealstategdp logroadmileage"

.
. //Other controls are not the same for both registrations and VMT
. //Registration Controls
. local spec2RegControls = "`spec1Controls' i.transactionDataDummy"

. local spec3RegControls = "`spec2RegControls' i.regDummyExtSpec3"

. local spec4RegControls = "`spec2RegControls' i.regDummyExtSpec4"

.
.
. di "`spec3RegControls'"
c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
> p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
> seddrivers logrealstategdp logroadmileage i.transactionDataDummy i.regDummy
> ExtSpec3

.
. //VMT Controls
. local spec2VMTControls = "`spec1Controls'"

. local spec3VMTControls = "`spec2VMTControls' i.vmtDummyExtSpec3"

. local spec4VMTControls = "`spec2VMTControls' i.vmtDummyExtSpec4"

.
. //Absorb Vars are always time and state fixed effects
. local absorbVars = "ib51.stateGroup ib1995.year"
```

```
498 .
499 . //Establish Cluster Variable
500 . local clusterVar = "stateGroup year"

501 .
502 . //Establish primary null
503 . local primaryNull = 1.6

504 . local altNull = 1

505 . local altNull2 = 0.3

506 . local altNull3 = 0

507 . local altNull4 = -2.5

508 .
509 . frame create ruralUrban

510 . frame change ruralUrban

511 .
512 . if("`c(os)'"=="Windows"){
513 .         local dataFile = "CleanedData\UrbanRuralVMT.csv"
514 . }

515 . else{
516 .         local dataFile = "CleanedData/UrbanRuralVMT.csv"
517 . }

518 .
519 . import delimited using "`dataFile'"
    (encoding automatically selected: ISO-8859-1)
    (4 vars, 2,652 obs)
```

```
520 .
521 . frame change default

522 . frlink 1:1 year state, frame(ruralUrban)
      (all observations in frame default matched)

523 . frget ruralvmt urbanvmt, from(ruralUrban)
      (2 variables copied from linked frame)

524 . gen logruralvmt = log(ruralvmt)
    (47 missing values generated)

525 . gen logurbanvmt = log(urbanvmt)

526 .
527 . bro state year ruralvmt urbanvmt

528 .
529 . //Urban model
530 . //non-normalized models
531 . local treatmentVar = "1.nosafetyind"

532 .
533 . local fsVarsSpec2 = "logregistrations i.nosafetyind `spec2RegControls'"

534 . local rfVarsSpec2 = "logurbanvmt i.nosafetyind `spec2VMTControls'"

535 . local modelTitleSpec2 = "Spec2Urban"

536 .
537 . local fsOut = "logregistrations"

538 . local rfOut = "logurbanvmt"
```

```
539 .
540 . local treatControl = "nosafetyind"

541 .
542 . local fsControlsSpec2 = "`spec2RegControls'"

543 . local rfControlsSpec2 = "`spec2VMTControls'"

544 .
545 .
546 . local nullLowSpec2 = -4

547 . local nullDeltaSpec2 = 0.001

548 . local nullHighSpec2 = 10

549 .
550 . // runModel `fsVarsSpec2', `rfVarsSpec2', `clusterVar', `treatmentVar', `mo
    > delTitleSpec2', `nullLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primary
    > Null', `altNull', `altNull2', `altNull3', `altNull4'
551 . runModelFWL `fsOut', `rfOut', `fsControlsSpec2', `rfControlsSpec2', `treatC
    > ontrol', `absorbVars', `clusterVar', `treatmentVar', `modelTitleSpec2', `nu
    > llLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primaryNull', `altNull', `
    > altNull2', `altNull3', `altNull4'
    logregistrations
    logurbanvmt
    c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
    > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
    > seddrivers logrealstategdp logroadmileage i.transactionDataDummy
    c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
    > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
    > seddrivers logrealstategdp logroadmileage
    nosafetyind
    ib51.stateGroup ib1995.year
    Working on:Spec2Urban
```

```
      Source │       SS           df       MS      Number of obs   =     2,42
> 4
─────────────┼──────────────────────────────────   F(1, 2423)      =     32.7
> 8
       Model │  .075299639         1  .075299639   Prob > F        =    0.000
> 0
    Residual │  5.56653287     2,423  .002297372   R-squared       =    0.013
> 3
─────────────┼──────────────────────────────────   Adj R-squared   =    0.012
> 9
       Total │  5.64183251     2,424  .002327489   Root MSE        =    .0479
> 3

─────────────────────┼──────────────────────────────────────────────────────────
> ─────────
logregistrations~s │ Coefficient  Std. err.      t    P>|t|     [95% conf. in
> terval]
─────────────────────┼──────────────────────────────────────────────────────────
> ─────────
fs_nosafetyind_res │   .0525274    .009175     5.73   0.000     .0345358
> .070519
─────────────────────┼──────────────────────────────────────────────────────────

> ─────────
Current approach can handle two-way clustering. Higher dimensional clustering
>  will produce incorrect results.

scalars:
              r(covar) =  .0001219344201541
           r(covarLow) =  .0001219344201541
          r(covarHigh) =  .0001229438990137
           r(deltaVar) =  .0009336235884885
              r(piVar) =  .0002269831190162
   r(covarUncorrected) =  .0001140231089826
  r(deltaVarUncorrect
    ed)               =  .0008658800073283
   r(piVarUncorrected) =  .0002105132594982
              r(delta) =  .0027361678085669
                 r(pi) =  .0525273813782378
       r(rfCorrection) =  1.069383401680796
       r(fsCorrection) =  1.07823668474482
                  r(N) =  2424
             r(nClust) =  48
            r(nClust2) =  48
            r(nClust1) =  51
  r(covarUncorrectedR) =  1.00501914046e-06
  r(deltaVarUncorrect
```

```
        edR)              =  .0001493706983705
      r(piVarUncorrectedR) =  .000091730792029
      r(covarUncorrected2) =  -.0000130052746697
      r(deltaVarUncorrect
        ed2)              =  .0000993048009664
      r(piVarUncorrected2) =  .0001238976598154
      r(covarUncorrected1) =  .0001280334027928
      r(deltaVarUncorrect
        ed1)              =  .0009159459047324
      r(piVarUncorrected1) =  .0001783463917119
                   r(rfK) =  110
                   r(fsK) =  129

552 .
553 . //Rural model
554 . preserve

555 . drop if state=="Dist. of Col."
    (49 observations deleted)

556 .
557 . local fsVarsSpec2 = "logregistrations i.nosafetyind `spec2RegControls'"

558 . local rfVarsSpec2 = "logruralvmt i.nosafetyind `spec2VMTControls'"

559 .
560 . local rfOut = "logruralvmt"

561 .
562 . local modelTitleSpec2 = "Spec2Rural"

563 . local nullLowSpec2 = -10

564 . local nullDeltaSpec2 = .01
```

```
565 . local nullHighSpec2 = 50

566 .
567 . runModelFWL `fsOut', `rfOut', `fsControlsSpec2', `rfControlsSpec2', `treatC
    > ontrol', `absorbVars', `clusterVar', `treatmentVar', `modelTitleSpec2', `nu
    > llLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primaryNull', `altNull', `
    > altNull2', `altNull3', `altNull4'
    logregistrations
    logruralvmt
    c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
    > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
    > seddrivers logrealstategdp logroadmileage i.transactionDataDummy
    c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
    > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
    > seddrivers logrealstategdp logroadmileage
    nosafetyind
    ib51.stateGroup ib1995.year
    Working on:Spec2Rural

          Source │       SS           df       MS      Number of obs   =     2,37
    > 6
    ─────────────┼──────────────────────────────────   F(1, 2375)      =     25.7
    > 8
           Model │   .05554681         1   .05554681   Prob > F        =    0.000
    > 0
        Residual │  5.11743543     2,375   .00215471   R-squared       =    0.010
    > 7
    ─────────────┼──────────────────────────────────   Adj R-squared   =    0.010
    > 3
           Total │  5.17298224     2,376  .002177181   Root MSE        =    .0464
    > 2

    ─────────────────────┬──────────────────────────────────────────────────────────
    > ────────
    logregistrations~s │ Coefficient  Std. err.      t    P>|t|     [95% conf. in
    > terval]
    ─────────────────────┼──────────────────────────────────────────────────────────
    > ────────
    fs_nosafetyind_res │   .0464165   .0091419     5.08   0.000     .0284896    .
    > 0643435
    ─────────────────────┴──────────────────────────────────────────────────────────
    > ────────
    Current approach can handle two-way clustering. Higher dimensional clustering
    >  will produce incorrect results.
```

```
    scalars:
                  r(covar) =  -.0000615467961005
                r(covarLow) =  -.0000615467961005
               r(covarHigh) =  -.0000610311924295
                r(deltaVar) =  .0005327078416854
                   r(piVar) =  .0002212379564952
       r(covarUncorrected) =  -.0000570673770896
      r(deltaVarUncorrect
        ed)                 =  .0004939369911375
       r(piVarUncorrected) =  .0002051361027668
                  r(delta) =  -.0166833548444636
                     r(pi) =  .0464165245441927
          r(rfCorrection) =  1.069458516267027
          r(fsCorrection) =  1.078493514848207
                      r(N) =  2376
                 r(nClust) =  48
                r(nClust2) =  48
                r(nClust1) =  50
      r(covarUncorrectedR) =  -3.08434864634e-06
      r(deltaVarUncorrect
        edR)                =  .0001403832056664
      r(piVarUncorrectedR) =  .0000545168957183
      r(covarUncorrected2) =  -3.50445381332e-06
      r(deltaVarUncorrect
        ed2)                =  .0001358120130671
      r(piVarUncorrected2) =  .0000632642110365
      r(covarUncorrected1) =  -.0000566472719226
      r(deltaVarUncorrect
        ed1)                =  .0004985081837368
      r(piVarUncorrected1) =  .0001963887874486
                    r(rfK) =  108
                    r(fsK) =  127

568 . // runModel `fsVarsSpec2', `rfVarsSpec2', `clusterVar', `treatmentVar', `mo
    > delTitleSpec2', `nullLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primary
    > Null', `altNull', `altNull2', `altNull3', `altNull4'
```

```
. 
. restore

. 
. // python script "`pythonSaveDir'combineIV.py"
. //
. //
. // constructTable VMTPerCapitaiv; Fleet Travel Distance Per Capita; Spec1Pe
> rCapita Spec2PerCapita Spec3PerCapita Spec4PerCapita; `primaryNull'; `altNu
> ll'; `altNull2'; `altNull3'; `altNull4'
. // constructTable VMTiv; Fleet Travel Distance ; Spec1 Spec2 Spec3 Spec4; `
> primaryNull'; `altNull'; `altNull2'; `altNull3'; `altNull4'
. 
end of do-file

. 
. //Interacted treatment with metro population share
. do RegDiDMetro

. eststo clear

. 
. if("`c(os)'"=="Windows"){
.         local tableSaveDir = "Tables\DiD\"
.         local plotSaveDir = "Plots\"
.         local pythonSaveDir = "PythonScripts\"
. }

. else{
.         local tableSaveDir = "Tables/DiD/"
.         local plotSaveDir = "Plots/"
.         local pythonSaveDir = "PythonScripts/"
. }
```

```
593 .
594 . //With linear and quadratic time trends
595 . //Establish Control Variable Sets
596 . local spec1Controls = "c.normYear#i.stateGroup c.quadYear#i.stateGroup asin
    > hMetroPop asinhNonMetroPop asinhMetroInc asinhNonMetroInc logrealmeangaspri
    > ce logemployment loglicenseddrivers logrealstategdp logroadmileage"

597 . local spec2Controls = "`spec1Controls' i.transactionDataDummy"

598 . local spec3Controls = "`spec2Controls' i.regDummyExtSpec3"

599 . local spec4Controls = "`spec2Controls' i.regDummyExtSpec4"

600 .
601 . //Establish ci level
602 . local ciLevel = 95

603 .
604 . //Establish Cluster Variable
605 . local clusterVar = "stateGroup year"

606 .
607 . //Absorb Vars are always time and state fixed effects
608 . local absorbVars = "ib51.stateGroup ib1995.year"

609 .
610 . //Estimate Models
611 . //Estimate Primary Spec. 2 DiD Models
612 . regFWL logregistrations, nosafetyind, `spec2Controls', `absorbVars', `clust
    > erVar', _
    absorb: ib51.stateGroup ib1995.year

    scalars:
                       e(N) =  2424
                  e(df_m) =  1
                  e(df_r) =  2423
                     e(F) =  32.77642113692301
                    e(r2) =  .0133466633423204
                  e(rmse) =  .0479309108348405
                   e(mss) =  .0752996391765457
                   e(rss) =  5.566532873207353
                  e(r2_a) =  .0129394601493127
                    e(ll) =  3925.09216070153
                  e(rank) =  1
```

```
    macros:
                e(cmdline) : "regress logregistrations_res  nosafetyind_res, noco
    > n"
                 e(title) : "Linear regression"
              e(marginsok) : "XB default"
                    e(vce) : "ols"
                e(depvar) : "logregistrations_res"
                   e(cmd) : "regress"
             e(properties) : "b V"
                e(predict) : "regres_p"
                  e(model) : "ols"
              e(estat_cmd) : "regress_estat"

    matrices:
                      e(b) :  1 x 1
                      e(V) :  1 x 1
                   e(beta) :  1 x 1

    functions:
                 e(sample)
```

	mean nosafet~s	lnvar _cons
mean		
nosafetyin~s	**.000227**	
lnvar		
_cons	**.0006376**	**.0169832**

```
613 . matlist e(b)
```

	mean nosafet~s	lnvar _cons
y1	**.0525274**	**-6.075989**

```
614 . matlist e(V)
```

	mean	lnvar
	nosafet~s	_cons
mean		
nosafetyin~s	**.000227**	
lnvar		
_cons	**.0006376**	**.0169832**

```
615 . //Estimate Alternative Spec. 2 DiD Model
616 . regFWL logregistrations, nosafetyind metroIntTreatment, `spec2Controls', `a
    > bsorbVars', `clusterVar', _
    absorb: ib51.stateGroup ib1995.year

    scalars:
                       e(N) =  2424
                    e(df_m) =  2
                    e(df_r) =  2422
                       e(F) =  18.63824523175877
                      e(r2) =  .0151575028704852
                    e(rmse) =  .0478967907910025
                     e(mss) =  .0855160925012557
                     e(rss) =  5.556316419882643
                    e(r2_a) =  .014344255556588
                      e(ll) =  3927.318630661779
                    e(rank) =  2

    macros:
                e(cmdline) : "regress logregistrations_res  nosafetyind_res metro
    > IntTreatment_res, nocon"
                  e(title) : "Linear regression"
              e(marginsok) : "XB default"
                    e(vce) : "ols"
                e(depvar) : "logregistrations_res"
                    e(cmd) : "regress"
            e(properties) : "b V"
               e(predict) : "regres_p"
                 e(model) : "ols"
             e(estat_cmd) : "regress_estat"
```

matrices:

```
                  e(b) :  1 x 2
                  e(V) :  2 x 2
               e(beta) :  1 x 2
```

functions:

```
             e(sample)
```

	mean		lnvar
	nosafet~s	metroIn~s	_cons
mean			
nosafetyin~s	**.002443**		
metroIntTr~s	**-.0036377**	**.0059349**	
lnvar			
_cons	**-.0013726**	**.0027701**	**.016772**

```
617 . matlist e(b)
```

	mean		lnvar
	nosafet~s	metroIn~s	_cons
y1	**-.0077175**	**.0917635**	**-6.077414**

```
618 . matlist e(V)
```

	mean		lnvar
	nosafet~s	metroIn~s	_cons
mean			
nosafetyin~s	**.002443**		
metroIntTr~s	**-.0036377**	**.0059349**	
lnvar			
_cons	**-.0013726**	**.0027701**	**.016772**

```
619 . eststo spec2

620 . local df = e(N_clust)-1

621 . addRobustFStat "mean:nosafetyind_res"

    added scalar:
                 e(fStat) =  .0243793

622 . addTreat "mean:nosafetyind_res"
    hello
    5
    mean
    nosafetyind_res

     ( 1)  [mean]nosafetyind_res = 0

               chi2(  1) =    0.02
             Prob > chi2 =    0.8759
    hello2

    added macro:
                e(treatb) : "0-.008"

623 .
624 . test [mean]nosafetyind_res

     ( 1)  [mean]nosafetyind_res = 0

               chi2(  1) =    0.02
             Prob > chi2 =    0.8759

625 . test [mean]metroIntTreatment_res

     ( 1)  [mean]metroIntTreatment_res = 0

               chi2(  1) =    1.42
             Prob > chi2 =    0.2336
```

```
626 .
627 . estadd local dsc = "Documented"

    added macro:
                   e(dsc) : "Documented"

628 . estadd local groupFE = "Y"

    added macro:
                e(groupFE) : "Y"

629 . estadd local timeFE = "Y"

    added macro:
                 e(timeFE) : "Y"

630 . estadd local timeTrend = "Y"

    added macro:
              e(timeTrend) : "Y"

631 . estadd local quadTrend = "Y"

    added macro:
              e(quadTrend) : "Y"

632 . estadd local commonControls = "Y"

    added macro:
         e(commonControls) : "Y"

633 . estadd local modelNum = "(2)"

    added macro:
               e(modelNum) : "(2)"
```

```
634 .
635 .
636 . //Export Model Results to TeX Files
637 . //Establish Format
638 . local fmtDiD = `"drop(*) nocon label compress starlevels(* 0.1 ** 0.05 ***
    > 0.01) nodepvar nomtitle nonumbers not nostar noline"'

639 .
640 .
641 . // local noteString = "\hline \hline \end{tabular} \vspace{0.2cm} \begin{ta
    > blenotes} \begin{spacing}{0} \begin{footnotesize} \item \textbf{Notes:} Thi
    > s table presents difference-in-differences estimates of the impact of the r
    > emoval of safety inspections on the log of vehicle registrations.  The unit
    >  of observation is state-year. The four specifications pertaining to differ
    > ent levels of ``Data Source Controls'' (DSCs) and control specifications ar
    > e documented in Table \cref{ssec:Spec}. Standard errors, reported in parent
    > heses, are clustered by state. * p\textless 0.1, ** p\textless 0.05, *** p\
    > textless 0.01 \end{footnotesize} \end{spacing} \end{tablenotes} \end{table}
    > "
642 . //
643 . // //Export Table
644 . // esttab spec1 spec2 spec3 spec4 using "`tableSaveDir'RegDiD.tex", `fmtDiD
    > ' title("Diff-in-Diff Results \label{tab:RegDiD}") stats(modelNum treatb tr
    > eatse dsc groupFE timeFE timeTrend quadTrend commonControls fStat N r2, lab
    > els("\hline \hline \multicolumn{5}{c}{Dep. Var.: Log Number of Vehicle Regi
    > strations} \\ " "Removal of" "Safety Inspections" "\makecell[l]{Data Source
    >  \nextline Controls}" "State Fixed Effects" "Year Fixed Effects" "\makecell
    > [l]{State-Specific Linear \\ Time Trends}" "\makecell[l]{State-Specific Qua
    > dratic \\ Time Trends}" "Common Controls" "Robust F-Stat" "Number of Obs."
    > "R\textsuperscript{2}")fmt(%8.3f %8.3f %8.3f %8.3f  %8.3f  %8.3f  %8.3f  %8
    > .3f %8.3f %8.0fc %8.3f)) replace postfoot("`noteString'")
645 .
    end of do-file
```

```
646 .
647 . // Urban-Rural Reduced Form Comparison
648 . capture frame drop ruralUrban

649 . capture drop ruralUrban ruralvmt urbanvmt logruralvmt logurbanvmt

650 . do VMTRFRuralUrban

651 . eststo clear

652 .
653 .
654 . if("`c(os)'"=="Windows"){
655 .         local tableSaveDir = "ReviewResponseResults\"
656 .         local plotSaveDir = "ReviewResponseResults\"
657 .         local pythonSaveDir = "PythonScripts\"
658 . }

659 . else{
660 .         local tableSaveDir = "ReviewResponseResults/"
661 .         local plotSaveDir = "ReviewResponseResults/"
662 .         local pythonSaveDir = "PythonScripts/"
663 . }

664 .
665 . //Establish Control Variable Sets
666 . //Spec1 controls are the same for both registrations and VMT
667 . local spec1Controls = "c.normYear#i.stateGroup c.quadYear#i.stateGroup asin
    > hMetroPop asinhNonMetroPop asinhMetroInc asinhNonMetroInc logrealmeangaspri
    > ce logemployment loglicenseddrivers logrealstategdp logroadmileage"

668 .
669 . //Other controls are not the same for both registrations and VMT
670 . //Registration Controls
```

```
671 . local spec2RegControls = "`spec1Controls' i.transactionDataDummy"

672 . local spec3RegControls = "`spec2RegControls' i.regDummyExtSpec3"

673 . local spec4RegControls = "`spec2RegControls' i.regDummyExtSpec4"

674 .
675 .
676 . di "`spec3RegControls'"
    c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
    > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
    > seddrivers logrealstategdp logroadmileage i.transactionDataDummy i.regDummy
    > ExtSpec3

677 .
678 . //VMT Controls
679 . local spec2VMTControls = "`spec1Controls'"

680 . local spec3VMTControls = "`spec2VMTControls' i.vmtDummyExtSpec3"

681 . local spec4VMTControls = "`spec2VMTControls' i.vmtDummyExtSpec4"

682 .
683 . //Absorb Vars are always time and state fixed effects
684 . local absorbVars = "ib51.stateGroup ib1995.year"

685 .
686 . //Establish Cluster Variable
687 . local clusterVar = "stateGroup year"

688 .
689 . //Establish primary null
690 . local primaryNull = 1.6
```

```
691 . local altNull = 1

692 . local altNull2 = 0.3

693 . local altNull3 = 0

694 . local altNull4 = -2.5

695 .
696 . frame create ruralUrban

697 . frame change ruralUrban

698 .
699 . if("`c(os)'"=="Windows"){
700 .         local dataFile = "CleanedData\UrbanRuralVMT.csv"
701 . }

702 . else{
703 .         local dataFile = "CleanedData/UrbanRuralVMT.csv"
704 . }

705 .
706 . import delimited using "`dataFile'"
    (encoding automatically selected: ISO-8859-1)
    (4 vars, 2,652 obs)

707 .
708 . frame change default

709 . frlink 1:1 year state, frame(ruralUrban)
      (all observations in frame default matched)

710 . frget ruralvmt urbanvmt, from(ruralUrban)
      (2 variables copied from linked frame)
```

```
711 . gen logruralvmt = log(ruralvmt)
    (47 missing values generated)

712 . gen logurbanvmt = log(urbanvmt)

713 .
714 .
715 . //Run urban model to get appropriate N and K values.
716 . qui reghdfe logurbanvmt nosafetyind `spec2VMTControls', absorb(`absorbVars'
    > ) cluster(`clusterVar')

717 . local k_u = e(df_a)+e(df_m)

718 . local N_u = e(N)

719 . local r2_u = e(r2)

720 . local N_clust_u = e(N_clust)

721 . local correction_u = (`N_clust_u')/(`N_clust_u'-1)*(`N_u'-1)/(`N_u'-`k_u')

722 .
723 . //Run rural model to get appropriate N and K values.
724 . qui reghdfe logruralvmt nosafetyind `spec2VMTControls' if state!="Dist. of
    > Col.", absorb(`absorbVars') cluster(`clusterVar')

725 . local k_r = e(df_a)+e(df_m)

726 . local N_r = e(N)

727 . local r2_r = e(r2)

728 . local N_clust_r = e(N_clust)
```

```
729 . local correction_r = (`N_clust_r')/(`N_clust_r'-1)*(`N_r'-1)/(`N_r'-`k_r')

730 .
731 . //Partial urban model variables
732 . qui reghdfe logurbanvmt `spec2VMTControls', absorb(`absorbVars') cluster(`c
    > lusterVar') res(logurbanvmt_res)

733 . qui reghdfe nosafetyind `spec2VMTControls', absorb(`absorbVars') cluster(`c
    > lusterVar') res(nosafetyind_res_u)

734 .
735 . //Partial urban model variables
736 . qui reghdfe logruralvmt `spec2VMTControls' if state!="Dist. of Col.", absor
    > b(`absorbVars') cluster(`clusterVar') res(logruralvmt_res)

737 . qui reghdfe nosafetyind `spec2VMTControls' if state!="Dist. of Col.", absor
    > b(`absorbVars') cluster(`clusterVar') res(nosafetyind_res_r)

738 .
739 . //Estimate partialled urban model
740 . regress logurbanvmt_res nosafetyind_res_u, nocon

      Source |       SS           df       MS      Number of obs   =     2,42
    > 4
-------------+------------------------------   F(1, 2423)      =      0.0
    > 4
       Model |  .000219706         1  .000219706   Prob > F        =    0.844
    > 7
    Residual |  13.8769148     2,423  .005727163   R-squared       =    0.000
    > 0
-------------+------------------------------   Adj R-squared   =   -0.000
    > 4
       Total |  13.8771345     2,424   .00572489   Root MSE        =    .0756
    > 8

-------------------+----------------------------------------------------------
    > -------
  logurbanvmt_res | Coefficient  Std. err.      t    P>|t|     [95% conf. int
    > erval]
-------------------+----------------------------------------------------------
    > -------
nosafetyind_res_u |   .0027362   .0139698     0.20   0.845    -.0246579    .0
    > 301302
-------------------+----------------------------------------------------------
    > -------
```

```
741 . estimates store urban

742 .
743 . //Estimate partialled rural model
744 . regress logruralvmt_res nosafetyind_res_r if state!="Dist. of Col.", nocon

      Source │       SS           df       MS      Number of obs   =     2,37
    > 6
─────────────┼──────────────────────────────────   F(1, 2375)      =      1.4
    > 6
       Model │  .007729717         1  .007729717   Prob > F        =    0.227
    > 6
    Residual │  12.6026585     2,375  .005306383   R-squared       =    0.000
    > 6
─────────────┼──────────────────────────────────   Adj R-squared   =    0.000
    > 2
       Total │  12.6103882     2,376  .005307402   Root MSE        =    .0728
    > 4

─────────────────┬──────────────────────────────────────────────────────────────
    > ──────
  logruralvmt_res │ Coefficient  Std. err.      t    P>|t|     [95% conf. int
    > erval]
─────────────────┼──────────────────────────────────────────────────────────────
    > ──────
nosafetyind_res_r │  -.0166834    .013823    -1.21   0.228    -.0437897     .
    > 010423
─────────────────┴──────────────────────────────────────────────────────────────
    > ──────

745 . estimates store rural

746 .
747 . suest urban rural, vce(cluster stateGroup)

    Simultaneous results for urban, rural                  Number of obs = 2,42
    > 4
```

```
                                (Std. err. adjusted for 51 clusters in state
> Group)
------------------------------------------------------------------------------
> -------
             |               Robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. int
> erval]
-------------+----------------------------------------------------------------
> -------
urban_mean   |
nosafetyind_res_u |   .0027362   .0302646     0.09   0.928    -.0565814    .0
> 620537
-------------+----------------------------------------------------------------
> -------
urban_lnvar  |
       _cons |  -5.162535   .3934673   -13.12   0.000    -5.933717   -4.
> 391353
-------------+----------------------------------------------------------------
> -------
rural_mean   |
nosafetyind_res_r |  -.0166834   .0223228    -0.75   0.455    -.0604353    .0
> 270686
-------------+----------------------------------------------------------------
> -------
rural_lnvar  |
       _cons |  -5.238845   .2100234   -24.94   0.000    -5.650483   -4.
> 827207
------------------------------------------------------------------------------
> -------

748 . mat var1 = e(V)

749 . suest urban rural, vce(cluster year)

Simultaneous results for urban, rural                  Number of obs = 2,42
> 4
```

```
                                                (Std. err. adjusted for 48 clusters in
> year)
------------------------------------------------------------------------------
> -------
             |               Robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. int
> erval]
-------------+----------------------------------------------------------------
> -------
urban_mean   |
nosafetyind_res_u |   .0027362   .0099652     0.27   0.784    -.0167952    .0
> 222676
-------------+----------------------------------------------------------------
> -------
urban_lnvar  |
       _cons |  -5.162535    .133141   -38.77   0.000    -5.423487   -4.
> 901584
-------------+----------------------------------------------------------------
> -------
rural_mean   |
nosafetyind_res_r |  -.0166834   .0116538    -1.43   0.152    -.0395245    .0
> 061578
-------------+----------------------------------------------------------------
> -------
rural_lnvar  |
       _cons |  -5.238845   .1074508   -48.76   0.000    -5.449445   -5.
> 028245
------------------------------------------------------------------------------
> -------

750 . mat var2 = e(V)

751 . suest urban rural, vce(robust)

Simultaneous results for urban, rural                  Number of obs = 2,42
> 4
```

```
------------------------------------------------------------------------------
> -------
                    |               Robust
                    | Coefficient  std. err.      z    P>|z|     [95% conf. int
> erval]
--------------------+---------------------------------------------------------
> -------
urban_mean          |
nosafetyind_res_u   |   .0027362   .0122217     0.22   0.823     -.021218    .0
> 266903
--------------------+---------------------------------------------------------
> -------
urban_lnvar         |
              _cons |  -5.162535   .1099934   -46.93   0.000    -5.378118   -4.
> 946952
--------------------+---------------------------------------------------------
> -------
rural_mean          |
nosafetyind_res_r   |  -.0166834   .0118483    -1.41   0.159    -.0399056    .0
> 065389
--------------------+---------------------------------------------------------
> -------
rural_lnvar         |
              _cons |  -5.238845   .0789903   -66.32   0.000    -5.393663   -5.
> 084027
--------------------+---------------------------------------------------------
> -------
```

```
752 . mat mat_beta = e(b)

753 . mat varR = e(V)

754 .
755 . mat mat_var = var1+var2-varR
```

```
. 
. matlist mat_beta
```

	urban_m~n nosafet~u	urban_l~r _cons	rural_m~n nosafet~r	rural_l~r _cons
y1	**.0027362**	**-5.162535**	**-.0166834**	**-5.238845**

```
. matlist mat_var
```

	urban_m~n nosafet~u	urban_l~r _cons	rural_m~n nosafet~r	rural_l~r _cons
urban_mean nosafetyin~u	**.0008659**			
urban_lnvar _cons	**.0037833**	**.1604445**		
rural_mean nosafetyin~r	**-.0000512**	**.0023294**	**.0004937**	
rural_lnvar _cons	**.0015294**	**.0574946**	**.0006068**	**.0494161**

```
. 
. mat diff_r = [1, 0, -1, 0]

. matlist diff_r
```

	c1	c2	c3	c4
r1	**1**	**0**	**-1**	**0**

```
762 .
763 . mat diff_value = diff_r*(mat_beta')

764 . matlist diff_value

             |        y1
-------------+----------
          r1 |  .0194195

765 .
766 . mat diff_var = diff_r*mat_var*(diff_r')

767 . matlist diff_var

             |        r1
-------------+----------
          r1 |   .001462

768 .
769 . local chi_stat = (diff_value[1,1]^2)/sqrt(diff_var[1,1])

770 .
771 . di "`chi_stat'"
    .009862956701897

772 .
773 . local p_value = 1-chi2(1, `chi_stat')

774 .
775 . di "`p_value'"
    .9208902176923863

776 .
    end of do-file
```

```
777 .
778 . //Emissions Controls
779 . do RegDIDEmissions

780 . eststo clear

781 .
782 .
783 . if("`c(os)'"=="Windows"){
784 .         local tableSaveDir = "ReviewResponseResults\"
785 .         local plotSaveDir = "ReviewResponseResults\"
786 .         local pythonSaveDir = "PythonScripts\"
787 . }

788 . else{
789 .         local tableSaveDir = "ReviewResponseResults/"
790 .         local plotSaveDir = "ReviewResponseResults/"
791 .         local pythonSaveDir = "PythonScripts/"
792 . }

793 .
794 . //Establish Control Variable Sets
795 . //Spec1 controls are the same for both registrations and VMT
796 . local spec1Controls = "c.normYear#i.stateGroup c.quadYear#i.stateGroup asin
    > hMetroPop asinhNonMetroPop asinhMetroInc asinhNonMetroInc logrealmeangaspri
    > ce logemployment loglicenseddrivers logrealstategdp logroadmileage i.emissi
    > onind"

797 .
798 . //Other controls are not the same for both registrations and VMT
799 . //Registration Controls
800 . local spec2Controls = "`spec1Controls' i.transactionDataDummy"

801 . local spec3Controls = "`spec2Controls' i.regDummyExtSpec3"
```

```
. local spec4Controls = "`spec2Controls' i.regDummyExtSpec4"

.
. //Absorb Vars are always time and state fixed effects
. local absorbVars = "ib51.stateGroup ib1995.year"

.
. //Establish Cluster Variable
. local clusterVar = "stateGroup year"

.
. //Establish primary null
. local primaryNull = 1.6

. local altNull = 1

. local altNull2 = 0.3

. local altNull3 = 0

. local altNull4 = -2.5

.
. frame create emissions

. frame change emissions

.
. if("`c(os)'"=="Windows"){
.         local dataFile = "CleanedData\EmissionsProgramsV2.xlsx"
. }

. else{
.         local dataFile = "CleanedData/EmissionsProgramsV2.xlsx"
```

```
825 . }

826 .
827 . import excel using "`dataFile'", first sheet("Clean") case(l)
    (4 vars, 51 obs)

828 .
829 . frame change default

830 . frlink m:1 state, frame(emissions)
      (all observations in frame default matched)

831 . frget stateemissionsprogram startemission endemission, from(emissions)
    (733 missing values generated)
    (733 missing values generated)
      (3 variables copied from linked frame)

832 .
833 . gen emissionind = 0

834 . replace emissionind = 1 if year>=startemission & year<=endemission & statee
    > missionsprogram==1
    (1,090 real changes made)

835 .
836 . frame drop emissions

837 .
838 . //non-normalized models
839 . local treatmentVar = "1.nosafetyind"

840 .
841 .
842 . local fsVarsSpec2 = "logregistrations i.nosafetyind `spec2RegControls' i.em
    > issionind"
```

```
843 . local rfVarsSpec2 = "logvmt i.nosafetyind `spec2VMTControls' i.emissionind"

844 .
845 .
846 . //Estimate Models
847 . //Estimate Spec. 1 DiD Model
848 . regFWL logregistrations, nosafetyind, `spec1Controls', `absorbVars', `clust
    > erVar', spec1ResEmissions
    absorb: ib51.stateGroup ib1995.year

    scalars:
                       e(N) =  2424
                    e(df_m) =  1
                    e(df_r) =  2423
                       e(F) =  33.04974310923899
                      e(r2) =  .0134564632503736
                    e(rmse) =  .0501185903003781
                     e(mss) =  .0830167604697003
                     e(rss) =  6.086268506028201
                    e(r2_a) =  .0130493053730523
                      e(ll) =  3816.905679308549
                    e(rank) =  1

    macros:
                 e(cmdline) : "regress logregistrations_res  nosafetyind_res, noco
    > n"
                   e(title) : "Linear regression"
              e(marginsok) : "XB default"
                    e(vce) : "ols"
                e(depvar) : "logregistrations_res"
                    e(cmd) : "regress"
              e(properties) : "b V"
                 e(predict) : "regres_p"
                   e(model) : "ols"
               e(estat_cmd) : "regress_estat"

    matrices:
                      e(b) :  1 x 1
                      e(V) :  1 x 1
                   e(beta) :  1 x 1

    functions:
                 e(sample)
```

```
                │ mean        │ lnvar
                │ nosafet~s   │     _cons
────────────────┼─────────────┼──────────
mean            │             │
nosafetyin~s    │  .0002154   │
────────────────┼─────────────┼──────────
lnvar           │             │
       _cons    │  .0006707   │  .0169904
```

```
849 . eststo spec1

850 . local df = e(N_clust)-1

851 . addRobustFStat "mean:nosafetyind_res"

    added scalar:
                 e(fStat) =  13.163028

852 . addTreat "mean:nosafetyind_res"
    hello
    5
    mean
    nosafetyind_res

     ( 1)  [mean]nosafetyind_res = 0

               chi2(  1) =   13.16
             Prob > chi2 =    0.0003
    hello2

    added macro:
                e(treatb) : "0.053***"

853 . estadd local dsc = "None"

    added macro:
                   e(dsc) : "None"
```

```
854 . estadd local groupFE = "Y"

    added macro:
                e(groupFE) : "Y"

855 . estadd local timeFE = "Y"

    added macro:
                 e(timeFE) : "Y"

856 . estadd local timeTrend = "Y"

    added macro:
              e(timeTrend) : "Y"

857 . estadd local quadTrend = "Y"

    added macro:
              e(quadTrend) : "Y"

858 . estadd local commonControls = "Y"

    added macro:
         e(commonControls) : "Y"

859 . estadd local emissionsControls = "Y"

    added macro:
      e(emissionsControls) : "Y"

860 . estadd local modelNum = "(1)"

    added macro:
               e(modelNum) : "(1)"
```

```
861 .
862 . //Estimate Spec. 2 DiD Model
863 . di "absorb og: `absorbVars'"
    absorb og: ib51.stateGroup ib1995.year

864 . regFWL logregistrations, nosafetyind, `spec2Controls', `absorbVars', `clust
    > erVar', spec2ResEmissions
    absorb: ib51.stateGroup ib1995.year

    scalars:
                       e(N) =  2424
                    e(df_m) =  1
                    e(df_r) =  2423
                       e(F) =  35.30336492098264
                      e(r2) =  .0143608658820338
                    e(rmse) =  .0477471919630849
                     e(mss) =  .0804844115425087
                     e(rss) =  5.523941686691509
                    e(r2_a) =  .0139540812620923
                      e(ll) =  3934.401189192723
                    e(rank) =  1

    macros:
                 e(cmdline) : "regress logregistrations_res  nosafetyind_res, noco
    > n"
                   e(title) : "Linear regression"
              e(marginsok) : "XB default"
                    e(vce) : "ols"
                  e(depvar) : "logregistrations_res"
                    e(cmd) : "regress"
              e(properties) : "b V"
                 e(predict) : "regres_p"
                   e(model) : "ols"
               e(estat_cmd) : "regress_estat"

    matrices:
                       e(b) :  1 x 1
                       e(V) :  1 x 1
                    e(beta) :  1 x 1

    functions:
                  e(sample)
```

```
                │ mean        │ lnvar
                │ nosafet~s   │     _cons
────────────────┼─────────────┼──────────
mean            │             │
nosafetyin~s    │  .0002275   │
────────────────┼─────────────┼──────────
lnvar           │             │
       _cons    │  .0006156   │  .0171707
```

```
865 . eststo spec2

866 . local df = e(N_clust)-1

867 . addRobustFStat "mean:nosafetyind_res"

    added scalar:
                 e(fStat) =  12.9888

868 . addTreat "mean:nosafetyind_res"
    hello
    5
    mean
    nosafetyind_res

     ( 1)  [mean]nosafetyind_res = 0

               chi2(  1) =   12.99
             Prob > chi2 =    0.0003
    hello2

    added macro:
                e(treatb) : "0.054***"

869 .
```

```
870 . estadd local dsc = "Documented"

    added macro:
                    e(dsc) : "Documented"

871 . estadd local groupFE = "Y"

    added macro:
                e(groupFE) : "Y"

872 . estadd local timeFE = "Y"

    added macro:
                 e(timeFE) : "Y"

873 . estadd local timeTrend = "Y"

    added macro:
              e(timeTrend) : "Y"

874 . estadd local quadTrend = "Y"

    added macro:
              e(quadTrend) : "Y"

875 . estadd local commonControls = "Y"

    added macro:
         e(commonControls) : "Y"

876 . estadd local emissionsControls = "Y"

    added macro:
      e(emissionsControls) : "Y"
```

```
877 . estadd local modelNum = "(2)"

    added macro:
               e(modelNum) : "(2)"

878 .
879 . //Estimate Spec. 3 DiD Model
880 . regFWL logregistrations, nosafetyind, `spec3Controls', `absorbVars', `clust
    > erVar', spec3ResEmissions
    absorb: ib51.stateGroup ib1995.year

    scalars:
                      e(N) =  2424
                   e(df_m) =  1
                   e(df_r) =  2423
                      e(F) =  17.96075660862686
                     e(r2) =  .0073580685637817
                   e(rmse) =  .0422514232088242
                    e(mss) =  .0320632331112334
                    e(rss) =  4.325497835163749
                   e(r2_a) =  .0069483938087523
                     e(ll) =  4230.813289236297
                   e(rank) =  1

    macros:
                e(cmdline) : "regress logregistrations_res  nosafetyind_res, noco
    > n"
                  e(title) : "Linear regression"
              e(marginsok) : "XB default"
                    e(vce) : "ols"
                e(depvar) : "logregistrations_res"
                    e(cmd) : "regress"
             e(properties) : "b V"
                e(predict) : "regres_p"
                  e(model) : "ols"
              e(estat_cmd) : "regress_estat"

    matrices:
                      e(b) :  1 x 1
                      e(V) :  1 x 1
                   e(beta) :  1 x 1

    functions:
                e(sample)
```

```
                │ mean        │ lnvar
                │ nosafet~s   │     _cons
────────────────┼─────────────┼──────────
mean            │             │
nosafetyin~s    │  .0001924   │
────────────────┼─────────────┼──────────
lnvar           │             │
        _cons   │  .0002135   │  .0154505
```

```
881 . eststo spec3

882 . local df = e(N_clust)-1

883 . addRobustFStat "mean:nosafetyind_res"

    added scalar:
                 e(fStat) =  6.2512591

884 . addTreat "mean:nosafetyind_res"
    hello
    5
    mean
    nosafetyind_res

     ( 1)  [mean]nosafetyind_res = 0

               chi2(  1) =    6.25
             Prob > chi2 =    0.0124
    hello2

    added macro:
                e(treatb) : "0.035**"

885 . estadd local dsc = "\makecell{Large \nextline Undocumented}"

    added macro:
                   e(dsc) : "\makecell{Large \nextline Undocumented}"
```

```
886 . estadd local groupFE = "Y"

    added macro:
                e(groupFE) : "Y"

887 . estadd local timeFE = "Y"

    added macro:
                 e(timeFE) : "Y"

888 . estadd local timeTrend = "Y"

    added macro:
              e(timeTrend) : "Y"

889 . estadd local quadTrend = "Y"

    added macro:
              e(quadTrend) : "Y"

890 . estadd local commonControls = "Y"

    added macro:
         e(commonControls) : "Y"

891 . estadd local emissionsControls = "Y"

    added macro:
      e(emissionsControls) : "Y"

892 . estadd local modelNum = "(3)"

    added macro:
               e(modelNum) : "(3)"
```

```
893 .
894 . //Estimate Spec. 4 Model
895 . regFWL logregistrations, nosafetyind, `spec4Controls', `absorbVars', `clust
    > erVar', spec4ResEmissions
    absorb: ib51.stateGroup ib1995.year

    scalars:
                       e(N) =  2424
                    e(df_m) =  1
                    e(df_r) =  2423
                       e(F) =  11.56766521758918
                      e(r2) =  .0047514248147034
                    e(rmse) =  .03602341290168
                     e(mss) =  .0150112004107781
                     e(rss) =  3.144293849376773
                    e(r2_a) =  .0043406742677842
                      e(ll) =  4617.366031810195
                    e(rank) =  1

    macros:
                 e(cmdline) : "regress logregistrations_res  nosafetyind_res, noco
    > n"
                   e(title) : "Linear regression"
              e(marginsok) : "XB default"
                    e(vce) : "ols"
                  e(depvar) : "logregistrations_res"
                    e(cmd) : "regress"
              e(properties) : "b V"
                 e(predict) : "regres_p"
                   e(model) : "ols"
               e(estat_cmd) : "regress_estat"

    matrices:
                       e(b) :  1 x 1
                       e(V) :  1 x 1
                    e(beta) :  1 x 1

    functions:
                  e(sample)
```

```
                │ mean        │ lnvar
                │ nosafet~s   │     _cons
────────────────┼─────────────┼──────────
mean            │             │
nosafetyin~s    │  .0001785   │
────────────────┼─────────────┼──────────
lnvar           │             │
       _cons    │ -.0000197   │  .0214171
```

```
896 . eststo spec4

897 . local df = e(N_clust)-1

898 . addRobustFStat "mean:nosafetyind_res"

    added scalar:
                 e(fStat) =  3.2949239

899 . addTreat "mean:nosafetyind_res"
    hello
    5
    mean
    nosafetyind_res

     ( 1)  [mean]nosafetyind_res = 0

               chi2(  1) =    3.29
             Prob > chi2 =    0.0695
    hello2

    added macro:
                e(treatb) : "0.024*"

900 . estadd local dsc = "\makecell{Small \nextline Undocumented}"

    added macro:
                   e(dsc) : "\makecell{Small \nextline Undocumented}"
```

```
901 . estadd local groupFE = "Y"

    added macro:
                e(groupFE) : "Y"

902 . estadd local timeFE = "Y"

    added macro:
                 e(timeFE) : "Y"

903 . estadd local timeTrend = "Y"

    added macro:
              e(timeTrend) : "Y"

904 . estadd local quadTrend = "Y"

    added macro:
              e(quadTrend) : "Y"

905 . estadd local commonControls = "Y"

    added macro:
         e(commonControls) : "Y"

906 . estadd local emissionsControls = "Y"

    added macro:
      e(emissionsControls) : "Y"

907 . estadd local modelNum = "(4)"

    added macro:
               e(modelNum) : "(4)"
```

```
908 .
909 . //Export Model Results to TeX Files
910 . //Establish Format
911 . local fmtDiD = `"drop(*) nocon label compress starlevels(* 0.1 ** 0.05 ***
    > 0.01) nodepvar nomtitle nonumbers not nostar noline"'

912 .
913 .
914 . local noteString = "\hline \hline \end{tabular} \vspace{0.2cm} \begin{table
    > notes} \begin{spacing}{0} \begin{footnotesize} \item \textbf{Notes:} This t
    > able presents difference-in-differences estimates of the impact of the remo
    > val of safety inspections on the log of vehicle registrations.  The unit of
    >  observation is state-year. The four specifications pertaining to different
    >  levels of ``Data Source Controls'' (DSCs) and control specifications are d
    > ocumented in Table \cref{ssec:Spec}. Standard errors, reported in parenthes
    > es, are clustered by state. * p\textless 0.1, ** p\textless 0.05, *** p\tex
    > tless 0.01 \end{footnotesize} \end{spacing} \end{tablenotes} \end{table}"

915 .
916 . //Export Table
917 . esttab spec1 spec2 spec3 spec4 using "`tableSaveDir'RegDiDEmissions.tex", `
    > fmtDiD' title("Diff-in-Diff Results Controlling for Emissions Inspections P
    > rograms \label{tab:RegDiDEmissions}") stats(modelNum treatb treatse dsc gro
    > upFE timeFE timeTrend quadTrend commonControls emissionsControls fStat N r2
    > , labels("\hline \hline \multicolumn{5}{c}{Dep. Var.: Log Number of Vehicle
    >  Registrations} \\ " "Removal of" "Safety Inspections" "\makecell[l]{Data S
    > ource \nextline Controls}" "State Fixed Effects" "Year Fixed Effects" "\mak
    > ecell[l]{State-Specific Linear \\ Time Trends}" "\makecell[l]{State-Specifi
    > c Quadratic \\ Time Trends}" "Common Controls" "\makecell[l]{Emission Inspe
    > ctions \\ Controls}" "Robust F-Stat" "Number of Obs." "R\textsuperscript{2}
    > ")fmt(%8.3f %8.3f %8.3f %8.3f  %8.3f  %8.3f  %8.3f  %8.3f %8.3f %8.0fc %8.3
    > f)) replace postfoot("`noteString'")
    (output written to ReviewResponseResults/RegDiDEmissions.tex)
```

```
. 
. 
end of do-file

. do LoadDataVMT

. clear

. //Load Data for travel analysis
. 
. 
. 
. if("`c(os)'"=="Windows"){
.         local dataFile = "CleanedData\FHWACleanV4.csv"
. }

. else{
.         local dataFile = "CleanedData/FHWACleanV4.csv"
. }

. 
. import delimited using "`dataFile'"
(encoding automatically selected: ISO-8859-1)
(90 vars, 3,009 obs)

. 
. //Add State Codes to all states
. sort state

. egen tempStateCode = mode(statecode), by(state)

. replace statecode = tempStateCode
(357 real changes made)
```

```
940 . drop tempStateCode

941 .
942 . //Label Variables
943 . label var logregistrations "Log Total Registrations"

944 . label var logvmt "Log VMT"

945 . label var loglicenseddrivers "Log Licensed Drivers"

946 . label var logjointpopulation "Log Total Population"

947 . label var logrealmeangasprice "Log Mean Gas Price"

948 . label var logemployment  "Log Employment"

949 . label var logrealjointincome  "Log Total Income"

950 . label var logmetropop  "Log (1 + Metro Population)"

951 . label var lognonmetropop  "Log (1 + Non-Metro Population)"

952 . label var logmetrorealinc  "Log (1 + Metro Income)"

953 . label var lognonmetrorealinc  "Log (1 + Non-Metro Income)"

954 . label var logrealstategdp  "Log State GDP"

955 . label var logpopulation  "Log Population"

956 . label var logrealtotalincome  "Log Total Income"

957 . label var logroadmileage "Log Road Mileage"
```

```
958 . label var nosafetyind "Treatment"

959 .
960 .
961 . //Create Inverse Hyperbolic Sine Variables
962 . gen asinhMetroInc = asinh(realmetroincome2018m)
    (459 missing values generated)

963 . gen asinhNonMetroInc = asinh(realnonmetroincome2018m)
    (459 missing values generated)

964 .
965 . gen asinhMetroPop = asinh(metropopulation)
    (459 missing values generated)

966 . gen asinhNonMetroPop = asinh(nonmetropopulation)
    (459 missing values generated)

967 .
968 . //Per capita model
969 . //construct outcome variable
970 . gen logRegPerCapita = logregistrations-log(metropopulation + nonmetropopula
    > tion)
    (510 missing values generated)

971 . gen logVMTPerCapita = logvmt-log(metropopulation + nonmetropopulation)
    (510 missing values generated)

972 .
973 . //Label Asinh variables
974 . label var asinhMetroInc "Asinh(Metro Income)"

975 . label var asinhNonMetroInc "Asinh(Non-Metro Income)"
```

```
976 .
977 . label var asinhMetroPop "Asinh(Metro Population)"

978 . label var asinhNonMetroPop "Asinh(Non-Metro Population)"

979 .
980 . //Label treatment variables
981 .
982 . label define nosafetyind 0 "No Treatment" 1 "Treatment"

983 . label values nosafetyind nosafetyind

984 .
985 .
986 . //Generate State Dummies
987 . egen stateGroup = group(state)

988 .
989 . //Drop Datapoints that are replications of past data for registrations
990 . drop if statecode=="CO" & year==2006
    (1 observation deleted)

991 . drop if statecode=="IN" & (year==2006 | year==2007 | year==2009)
    (3 observations deleted)

992 . drop if statecode=="MT" & year==2005
    (1 observation deleted)

993 . drop if statecode=="NJ" & year==2008
    (1 observation deleted)

994 . drop if statecode=="PR" & (year==2002 | year==2003 | year==2005 | year==200
    > 8 | year==2009 | year==2010)
    (0 observations deleted)
```

```
 995 . drop if statecode=="TX" & year==2009
     (1 observation deleted)

 996 . drop if statecode=="IL" & year==2011
     (1 observation deleted)

 997 . drop if statecode=="NH" & year==2012
     (1 observation deleted)

 998 . drop if statecode=="NY" & year==2012
     (1 observation deleted)

 999 . //Drop Datapoints that are replications of past data for VMT
1000 . drop if statecode=="MO" & year == 2003
     (1 observation deleted)

1001 . drop if statecode=="IN" & (year == 2004 | year==2009)
     (1 observation deleted)

1002 . drop if statecode=="NV" & year == 2004
     (1 observation deleted)

1003 . drop if statecode=="NH" & year == 2004
     (1 observation deleted)

1004 . drop if statecode=="NY" & year == 2005
     (1 observation deleted)

1005 . drop if statecode=="AZ" & year == 2009
     (1 observation deleted)

1006 . drop if statecode=="WY" & year == 2010
     (1 observation deleted)
```

```
1007 . drop if statecode=="PR" & (year == 2011 | year == 2012 | year == 2013 | yea
     > r == 2015 | year == 2016)
     (0 observations deleted)

1008 .
1009 . //Drop if data is prior to 1970
1010 . drop if year<1970
     (510 observations deleted)

1011 .
1012 . //Normalize differenece of outcome variables for identification of extreme
     > points
1013 .
1014 . sort stateGroup year

1015 . xtset stateGroup year

     Panel variable: stateGroup (unbalanced)
      Time variable: year, 1970 to 2018, but with gaps
              Delta: 1 unit

1016 .
1017 . gen DiffLogReg = d.logregistrations
     (118 missing values generated)

1018 . gen DiffLogVMT = d.logvmt
     (118 missing values generated)

1019 . gen DiffLogRegPerCapita = d.logRegPerCapita
     (118 missing values generated)

1020 . gen DiffLogVMTPerCapita = d.logVMTPerCapita
     (118 missing values generated)
```

```
1021 .
1022 . local normVarList = "DiffLogReg DiffLogVMT DiffLogRegPerCapita DiffLogVMTPe
     > rCapita"

1023 .
1024 . sort stateGroup

1025 .
1026 . foreach varToNorm in `normVarList'{
       2.
1027 .         by stateGroup: egen tempMean = mean(`varToNorm')
       3.         by stateGroup: egen tempSD = sd(`varToNorm')
       4.         by stateGroup: egen tempNumData = count(`varToNorm')
       5.         gen tempSE = tempSD/sqrt(tempNumData)
       6.
1028 .         gen norm`varToNorm' = (`varToNorm'-tempMean)/tempSD
       7.
1029 .         drop tempMean tempSD tempNumData tempSE
       8.
1030 .         di "`varToNorm'"
       9.
1031 . }
     (118 missing values generated)
     DiffLogReg
     (118 missing values generated)
     DiffLogVMT
     (118 missing values generated)
     DiffLogRegPerCapita
     (118 missing values generated)
     DiffLogVMTPerCapita

1032 .
1033 . //Generate Colorado Dummy
1034 . gen coDum1 = 0
```

```
1035 . gen coDum2 = 0

1036 . replace coDum1 = 1 if statecode=="CO" & year>=2002 & year<=2009
     (7 real changes made)

1037 . replace coDum2 = 1 if statecode=="CO" & year>=2010
     (9 real changes made)

1038 .
1039 . drop if coDum1 == 1
     (7 observations deleted)

1040 .
1041 .
1042 . //Generate dummy variable for data footnotes
1043 . gen transactionDataDummy = 0

1044 .
1045 . replace transactionDataDummy = 1 if statecode=="AR" & year>=2011
     (8 real changes made)

1046 . replace transactionDataDummy = 2 if statecode=="GA" & year>=2011
     (8 real changes made)

1047 . replace transactionDataDummy = 3 if statecode=="IA" & year>=2011
     (8 real changes made)

1048 . replace transactionDataDummy = 4 if statecode=="IL" & year>=2012
     (7 real changes made)

1049 . replace transactionDataDummy = 5 if statecode=="KY" & year>=2011
     (8 real changes made)

1050 . replace transactionDataDummy = 6 if statecode=="LA" & year>=2012
     (7 real changes made)
```

```
1051 . replace transactionDataDummy = 7 if statecode=="ME" & year>=2011 //Maine is
     >  a state with what looks to not have a consistent change
     (8 real changes made)

1052 . replace transactionDataDummy = 8 if statecode=="ME" & year>2014
     (4 real changes made)

1053 . replace transactionDataDummy = 9 if statecode=="MI" & year>=2011 //Michigan
     >  is a state with what looks to not have a consistent change
     (8 real changes made)

1054 . replace transactionDataDummy = 10 if statecode=="MI" & year>2012
     (6 real changes made)

1055 . replace transactionDataDummy = 11 if statecode=="MN" & year>=2011
     (8 real changes made)

1056 . replace transactionDataDummy = 12 if statecode=="NV" & year>=2011
     (8 real changes made)

1057 . replace transactionDataDummy = 13 if statecode=="OK" & year>=2011
     (8 real changes made)

1058 . replace transactionDataDummy = 14 if statecode=="SD" & year>=2011
     (8 real changes made)

1059 . replace transactionDataDummy = 15 if statecode=="TN" & year>=2011
     (8 real changes made)

1060 . replace transactionDataDummy = 16 if statecode=="TX" & year>=2011
     (8 real changes made)

1061 . replace transactionDataDummy = 17 if statecode=="WA" & year>=2011
     (8 real changes made)
```

```
1062 . replace transactionDataDummy = 18 if statecode=="WI" & year>=2011
     (8 real changes made)

1063 . replace transactionDataDummy = 19 if statecode=="WY" & year>=2011
     (8 real changes made)

1064 .
1065 .
1066 .
1067 .
1068 . //Local Extreme Discontinuity controls
1069 .
1070 . sort state year

1071 .
1072 . python
     ---------------------------------------------------- python (type end to exit) ---
     >>>
     >>> import numpy as np
     >>> import pandas as p
     >>> from sfi import Data
     >>>
     >>>
     >>> def getDSCDummies(data, col, cutoff):
     ...
     ...         tempExtState = None
     ...
     ...         tempIndex = 0
     ...         stateCol = 0
     ...
     ...         rows, cols = data.shape
     ...
     ...         dummyExt = []
     ...
     ...         for i in range(rows):
     ...                 tempDiff = float(data[i,col])
     ...                 tempState = data[i,stateCol]
     ...
     ...
     ...                 if(abs(tempDiff)>cutoff):
     ...                         tempExtState = tempState
     ...                         tempIndex+=1
     ...
     ...                 if tempState==tempExtState:
     ...                         dummyExt.append(tempIndex)
     ...                 else:
```

```
...                         dummyExt.append(0)
...
...
...         dummyExt = np.array(dummyExt).reshape((-1,1))
...         return dummyExt
...
>>> dataCols = ['state', 'year', 'normDiffLogReg', 'normDiffLogVMT', 'normDif
> fLogRegPerCapita', 'normDiffLogVMTPerCapita']
>>>
>>> data = np.array(Data.get(dataCols, missingval=np.nan))
>>>
>>> print(data)
[['Alabama' '1970' 'nan' 'nan' 'nan' 'nan']
 ['Alabama' '1971' '1.0640671570008131' '0.3361620217656708'
  '0.9150116516585626' '0.16061868662892634']
 ['Alabama' '1972' '1.0231103264075507' '0.24914037544421433'
  '0.9109946118213749' '0.11750494756107976']
 ...
 ['Wyoming' '2016' '0.45763583139776387' '-1.6128834808565655'
  '0.7516965179523265' '-1.1845977355288309']
 ['Wyoming' '2017' '-1.1510098900675998' '0.6274459465980643'
  '-0.8234171123760192' '1.2249676800199891']
 ['Wyoming' '2018' 'nan' 'nan' 'nan' 'nan']]
>>>
>>> regCutoffLow = 2.93 #Spec. 4
>>> regCutoffPrimary = 4.02 #Spec. 3
>>> regCutoffHigh = 7 #Spec. 2
>>>
>>> vmtCutoffLow = 2.93 #Spec. 4
>>> vmtCutoffPrimary = 4.02 #Spec. 3
>>> vmtCutoffHigh = 7 #Spec. 2
>>>
>>> regCol = dataCols.index('normDiffLogReg')
>>> vmtCol = dataCols.index('normDiffLogVMT')
>>>
>>> regPerCapitaCol = dataCols.index('normDiffLogRegPerCapita')
>>> vmtPerCapitaCol = dataCols.index('normDiffLogVMTPerCapita')
>>>
>>> rows, cols = data.shape
>>>
>>> regDummyExtLow = getDSCDummies(data, regCol, regCutoffLow)
>>> regDummyExtPrimary = getDSCDummies(data, regCol, regCutoffPrimary)
>>> regDummyExtHigh = getDSCDummies(data, regCol, regCutoffHigh)
>>>
>>> vmtDummyExtLow = getDSCDummies(data, vmtCol, vmtCutoffLow)
>>> vmtDummyExtPrimary = getDSCDummies(data, vmtCol, vmtCutoffPrimary)
```

```
>>> vmtDummyExtHigh = getDSCDummies(data, vmtCol, vmtCutoffHigh)
>>>
>>> regPerCapitaDummyExtLow = getDSCDummies(data, regPerCapitaCol, regCutoffL
> ow)
>>> regPerCapitaDummyExtPrimary = getDSCDummies(data, regPerCapitaCol, regCut
> offPrimary)
>>> regPerCapitaDummyExtHigh = getDSCDummies(data, regPerCapitaCol, regCutoff
> High)
>>>
>>> vmtPerCapitaDummyExtLow = getDSCDummies(data, vmtPerCapitaCol, vmtCutoffL
> ow)
>>> vmtPerCapitaDummyExtPrimary = getDSCDummies(data, vmtPerCapitaCol, vmtCut
> offPrimary)
>>> vmtPerCapitaDummyExtHigh = getDSCDummies(data, vmtPerCapitaCol, vmtCutoff
> High)
>>>
>>> Data.addVarInt('regDummyExtSpec4')
>>> Data.store("regDummyExtSpec4",None, regDummyExtLow[:])
>>> Data.addVarInt('regDummyExtSpec3')
>>> Data.store("regDummyExtSpec3",None, regDummyExtPrimary[:])
>>> Data.addVarInt('regDummyExtSpec2')
>>> Data.store("regDummyExtSpec2",None, regDummyExtHigh[:])
>>>
>>> Data.addVarInt('vmtDummyExtSpec4')
>>> Data.store("vmtDummyExtSpec4",None, vmtDummyExtLow[:])
>>> Data.addVarInt('vmtDummyExtSpec3')
>>> Data.store("vmtDummyExtSpec3",None, vmtDummyExtPrimary[:])
>>> Data.addVarInt('vmtDummyExtSpec2')
>>> Data.store("vmtDummyExtSpec2",None, vmtDummyExtHigh[:])
>>>
>>>
>>> Data.addVarInt('regPerCapitaDummyExtSpec4')
>>> Data.store("regPerCapitaDummyExtSpec4",None, regPerCapitaDummyExtLow[:])
>>> Data.addVarInt('regPerCapitaDummyExtSpec3')
>>> Data.store("regPerCapitaDummyExtSpec3",None, regPerCapitaDummyExtPrimary[
> :])
>>> Data.addVarInt('regPerCapitaDummyExtSpec2')
>>> Data.store("regPerCapitaDummyExtSpec2",None, regPerCapitaDummyExtHigh[:])
>>>
>>> Data.addVarInt('vmtPerCapitaDummyExtSpec4')
>>> Data.store("vmtPerCapitaDummyExtSpec4",None, vmtPerCapitaDummyExtLow[:])
>>> Data.addVarInt('vmtPerCapitaDummyExtSpec3')
>>> Data.store("vmtPerCapitaDummyExtSpec3",None, vmtPerCapitaDummyExtPrimary[
> :])
>>> Data.addVarInt('vmtPerCapitaDummyExtSpec2')
>>> Data.store("vmtPerCapitaDummyExtSpec2",None, vmtPerCapitaDummyExtHigh[:])
```

```
>>>
>>> end
```

```
1073 .
1074 .
1075 . //Generate normalized year values to reduce differences in magnitude of cov
     > ariates.
1076 . //Scale year for better numerical performance
1077 . local yearScale = 1

1078 . gen normYear = year/`yearScale'

1079 . gen quadYear = (year/1)^2

1080 .
1081 . //Establish treatment interacted with pre-treatment population share in met
     > ro areas
1082 . frame copy default MetroShares

1083 . frame change MetroShares

1084 . keep if year==1970
     (2,424 observations deleted)

1085 .
1086 . capture gen preTreatMetroShare = metropopulation/(metropopulation+nonmetrop
     > opulation)

1087 .
1088 . frame change default

1089 .
1090 . cap frlink m:1 state, frame(MetroShares)
```

```
1091 . cap frget preTreatMetroShare, from(MetroShares)

1092 .
1093 . cap gen metroIntTreatment = nosafetyind*preTreatMetroShare

1094 .
1095 . frame drop MetroShares

1096 .
     end of do-file

1097 . do VMTIVEmissions

1098 . eststo clear

1099 .
1100 .
1101 . if("`c(os)'"=="Windows"){
1102 .         local tableSaveDir = "ReviewResponseResults\"
1103 .         local plotSaveDir = "ReviewResponseResults\"
1104 .         local pythonSaveDir = "PythonScripts\"
1105 . }

1106 . else{
1107 .         local tableSaveDir = "ReviewResponseResults/"
1108 .         local plotSaveDir = "ReviewResponseResults/"
1109 .         local pythonSaveDir = "PythonScripts/"
1110 . }

1111 .
1112 . //Establish Control Variable Sets
1113 . //Spec1 controls are the same for both registrations and VMT
1114 . local spec1Controls = "c.normYear#i.stateGroup c.quadYear#i.stateGroup asin
     > hMetroPop asinhNonMetroPop asinhMetroInc asinhNonMetroInc logrealmeangaspri
     > ce logemployment loglicenseddrivers logrealstategdp logroadmileage"
```

```
. 
. //Other controls are not the same for both registrations and VMT
. //Registration Controls
. local spec2RegControls = "`spec1Controls' i.transactionDataDummy"

. local spec3RegControls = "`spec2RegControls' i.regDummyExtSpec3"

. local spec4RegControls = "`spec2RegControls' i.regDummyExtSpec4"

. 
. 
. di "`spec3RegControls'"
c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
> p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
> seddrivers logrealstategdp logroadmileage i.transactionDataDummy i.regDummy
> ExtSpec3

. 
. //VMT Controls
. local spec2VMTControls = "`spec1Controls'"

. local spec3VMTControls = "`spec2VMTControls' i.vmtDummyExtSpec3"

. local spec4VMTControls = "`spec2VMTControls' i.vmtDummyExtSpec4"

. 
. //Absorb Vars are always time and state fixed effects
. local absorbVars = "ib51.stateGroup ib1995.year"

. 
. //Establish Cluster Variable
. local clusterVar = "stateGroup year"

. 
```

```
. //Establish primary null
. local primaryNull = 1.6

. local altNull = 1

. local altNull2 = 0.3

. local altNull3 = 0

. local altNull4 = -2.5

.
. frame create emissions

. frame change emissions

.
. if("`c(os)'"=="Windows"){
.         local dataFile = "CleanedData\EmissionsProgramsV2.xlsx"
. }

. else{
.         local dataFile = "CleanedData/EmissionsProgramsV2.xlsx"
. }

.
. import excel using "`dataFile'", first sheet("Clean") case(l)
(4 vars, 51 obs)

.
. frame change default

. frlink m:1 state, frame(emissions)
  (all observations in frame default matched)
```

```
1157 . frget stateemissionsprogram startemission endemission, from(emissions)
     (733 missing values generated)
     (733 missing values generated)
       (3 variables copied from linked frame)

1158 .
1159 . gen emissionind = 0

1160 . replace emissionind = 1 if year>=startemission & year<=endemission & statee
     > missionsprogram==1
     (1,090 real changes made)

1161 .
1162 . frame drop emissions

1163 .
1164 . //non-normalized models
1165 . local treatmentVar = "1.nosafetyind"

1166 .
1167 .
1168 . local fsVarsSpec2 = "logregistrations i.nosafetyind `spec2RegControls' i.em
     > issionind"

1169 . local rfVarsSpec2 = "logvmt i.nosafetyind `spec2VMTControls' i.emissionind"

1170 .
1171 . local fsOut = "logregistrations"

1172 . local rfOut = "logvmt"

1173 .
1174 . local treatControl = "nosafetyind"
```

```
1175 .
1176 . local fsControlsSpec1 = "`spec1Controls' i.emissionind"

1177 . local rfControlsSpec1 = "`spec1Controls' i.emissionind"

1178 .
1179 .
1180 . local modelTitleSpec1 = "Spec1Emissions"

1181 . local nullLowSpec1 = -4

1182 . local nullDeltaSpec1 = 0.001

1183 . local nullHighSpec1 = 1

1184 .
1185 . runModelFWL `fsOut', `rfOut', `fsControlsSpec1', `rfControlsSpec1', `treatC
     > ontrol', `absorbVars', `clusterVar', `treatmentVar', `modelTitleSpec1', `nu
     > llLowSpec1', `nullDeltaSpec1', `nullHighSpec1', `primaryNull', `altNull', `
     > altNull2', `altNull3', `altNull4'
     logregistrations
     logvmt
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.emissionind
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.emissionind
     nosafetyind
     ib51.stateGroup ib1995.year
     Working on:Spec1Emissions

          Source │       SS           df       MS      Number of obs   =     2,42
     > 4
     ─────────────┼──────────────────────────────────   F(1, 2423)      =     33.0
     > 5
           Model │   .08301676         1   .08301676   Prob > F        =    0.000
     > 0
        Residual │  6.08626851     2,423  .002511873   R-squared       =    0.013
     > 5
     ─────────────┼──────────────────────────────────   Adj R-squared   =    0.013
     > 0
           Total │  6.16928527     2,424  .002545085   Root MSE        =    .0501
     > 2
```

```
------------------------------------------------------------------------------
> -------
logregistrations~s | Coefficient  Std. err.      t    P>|t|     [95% conf. in
> terval]
------------------------------------------------------------------------------
> -------
fs_nosafetyind_res |   .0532531   .0092632     5.75   0.000     .0350885    .
> 0714176
------------------------------------------------------------------------------
> -------
Current approach can handle two-way clustering. Higher dimensional clustering
>  will produce incorrect results.

scalars:
              r(covar) =  .0000259109536037
           r(covarLow) =  .0000259109536037
          r(covarHigh) =  .0000259109536037
           r(deltaVar) =  .0002845415852729
              r(piVar) =  .0002154433972943
   r(covarUncorrected) =  .0000242193361983
  r(deltaVarUncorrect
    ed)                =  .00026596505947
   r(piVarUncorrected) =  .0002013780021603
              r(delta) =  -.0096017744134962
                 r(pi) =  .0532530521703231
       r(rfCorrection) =  1.069845737781825
       r(fsCorrection) =  1.069845737781825
                  r(N) =  2424
             r(nClust) =  48
            r(nClust2) =  48
            r(nClust1) =  51
  r(covarUncorrectedR) =  2.01655271599e-06
  r(deltaVarUncorrect
    edR)               =  .0000396805449167
  r(piVarUncorrectedR) =  .0000903020694787
  r(covarUncorrected2) =  -1.85902072489e-06
  r(deltaVarUncorrect
    ed2)               =  .0000336002660839
  r(piVarUncorrected2) =  .000103736870988
  r(covarUncorrected1) =  .0000280949096392
  r(deltaVarUncorrect
    ed1)               =  .0002720453383028
  r(piVarUncorrected1) =  .000187943200651
               r(rfK) =  111
               r(fsK) =  111
```

```
1186 .
1187 . local fsControlsSpec2 = "`spec2RegControls' i.emissionind"

1188 . local rfControlsSpec2 = "`spec2VMTControls' i.emissionind"

1189 .
1190 .
1191 . local modelTitleSpec2 = "Spec2Emissions"

1192 . local nullLowSpec2 = -4

1193 . local nullDeltaSpec2 = 0.001

1194 . local nullHighSpec2 = 1

1195 .
1196 . runModelFWL `fsOut', `rfOut', `fsControlsSpec2', `rfControlsSpec2', `treatC
     > ontrol', `absorbVars', `clusterVar', `treatmentVar', `modelTitleSpec2', `nu
     > llLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primaryNull', `altNull', `
     > altNull2', `altNull3', `altNull4'
     logregistrations
     logvmt
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.transactionDataDummy i.emission
     > ind
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.emissionind
     nosafetyind
     ib51.stateGroup ib1995.year
     Working on:Spec2Emissions

          Source │       SS           df       MS      Number of obs   =     2,42
     > 4
     ────────────┼──────────────────────────────────   F(1, 2423)      =     35.3
     > 0
           Model │  .080484412         1  .080484412   Prob > F        =    0.000
     > 0
        Residual │  5.52394169     2,423  .002279794   R-squared       =    0.014
     > 4
     ────────────┼──────────────────────────────────   Adj R-squared   =    0.014
     > 0
           Total │   5.6044261     2,424  .002312057   Root MSE        =    .0477
     > 5
```

```
─────────────────────┬──────────────────────────────────────────────────────────
> ────────
logregistrations~s │ Coefficient  Std. err.      t    P>|t|     [95% conf. in
> terval]
─────────────────────┼──────────────────────────────────────────────────────────
> ────────
fs_nosafetyind_res │   .0543644   .0091497     5.94   0.000     .0364223    .
> 0723064
─────────────────────┴──────────────────────────────────────────────────────────
> ────────
Current approach can handle two-way clustering. Higher dimensional clustering
>  will produce incorrect results.

scalars:
              r(covar) =  .0000364039829429
           r(covarLow) =  .0000364039829429
          r(covarHigh) =  .0000367054980589
           r(deltaVar) =  .0002868982941308
              r(piVar) =  .0002275408524375
   r(covarUncorrected) =  .0000340273196942
  r(deltaVarUncorrect
    ed)               =  .00026596505947
   r(piVarUncorrected) =  .000210938571572
              r(delta) =  -.0096017744134962
                 r(pi) =  .054364351167051
       r(rfCorrection) =  1.069845737781825
       r(fsCorrection) =  1.078706709454822
                  r(N) =  2424
             r(nClust) =  48
            r(nClust2) =  48
            r(nClust1) =  51
  r(covarUncorrectedR) =  9.30950277249e-07
  r(deltaVarUncorrect
    edR)              =  .0000396805449167
  r(piVarUncorrectedR) =  .0000921437909359
  r(covarUncorrected2) =  -1.63084038450e-06
  r(deltaVarUncorrect
    ed2)              =  .0000336002660839
  r(piVarUncorrected2) =  .0001264189533276
  r(covarUncorrected1) =  .0000365891103559
  r(deltaVarUncorrect
    ed1)              =  .0002720453383028
  r(piVarUncorrected1) =  .0001766634091803
                r(rfK) =  111
                r(fsK) =  130
```

```
1197 . // runModel `fsVarsSpec2', `rfVarsSpec2', `clusterVar', `treatmentVar', `mo
     > delTitleSpec2', `nullLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primary
     > Null', `altNull', `altNull2', `altNull3', `altNull4'
1198 . //
1199 . // python script "`pythonSaveDir'combineIV.py"
1200 . //
1201 . //
1202 . // constructTable VMTPerCapitaiv; Fleet Travel Distance Per Capita; Spec1Pe
     > rCapita Spec2PerCapita Spec3PerCapita Spec4PerCapita; `primaryNull'; `altNu
     > ll'; `altNull2'; `altNull3'; `altNull4'
1203 . // constructTable VMTiv; Fleet Travel Distance ; Spec1 Spec2 Spec3 Spec4; `
     > primaryNull'; `altNull'; `altNull2'; `altNull3'; `altNull4'
1204 .
     end of do-file

1205 .
1206 . // Gas Use
1207 . do LoadDataGasUse

1208 . clear

1209 . //Load Data for gas use analysis
1210 . if("`c(os)'"=="Windows"){
1211 .         local dataFile = "CleanedData\FHWACleanV4.csv"
1212 . }

1213 . else{
1214 .         local dataFile = "CleanedData/FHWACleanV4.csv"
1215 . }

1216 .
1217 . import delimited using "`dataFile'"
     (encoding automatically selected: ISO-8859-1)
     (90 vars, 3,009 obs)
```

```
1218 .
1219 . //Add State Codes to all states
1220 . sort state

1221 . egen tempStateCode = mode(statecode), by(state)

1222 . replace statecode = tempStateCode
     (357 real changes made)

1223 . drop tempStateCode

1224 .
1225 . //Label Variables
1226 . label var logregistrations "Log Total Registrations"

1227 . label var logvmt "Log VMT"

1228 . label var loglicenseddrivers "Log Licensed Drivers"

1229 . label var logjointpopulation "Log Total Population"

1230 . label var logrealmeangasprice "Log Mean Gas Price"

1231 . label var logemployment  "Log Employment"

1232 . label var logrealjointincome  "Log Total Income"

1233 . label var logmetropop  "Log (1 + Metro Population)"

1234 . label var lognonmetropop  "Log (1 + Non-Metro Population)"

1235 . label var logmetrorealinc  "Log (1 + Metro Income)"

1236 . label var lognonmetrorealinc  "Log (1 + Non-Metro Income)"
```

```
1237 . label var logrealstategdp  "Log State GDP"

1238 . label var logpopulation  "Log Population"

1239 . label var logrealtotalincome  "Log Total Income"

1240 . label var logroadmileage "Log Road Mileage"

1241 . label var nosafetyind "Treatment"

1242 .
1243 .
1244 . //Create Inverse Hyperbolic Sine Variables
1245 . gen asinhMetroInc = asinh(realmetroincome2018m)
     (459 missing values generated)

1246 . gen asinhNonMetroInc = asinh(realnonmetroincome2018m)
     (459 missing values generated)

1247 .
1248 . gen asinhMetroPop = asinh(metropopulation)
     (459 missing values generated)

1249 . gen asinhNonMetroPop = asinh(nonmetropopulation)
     (459 missing values generated)

1250 .
1251 .
1252 . //Label Asinh variables
1253 . label var asinhMetroInc "Asinh(Metro Income)"

1254 . label var asinhNonMetroInc "Asinh(Non-Metro Income)"

1255 .
```

```
1256 . label var asinhMetroPop "Asinh(Metro Population)"

1257 . label var asinhNonMetroPop "Asinh(Non-Metro Population)"

1258 .
1259 . //per capita models
1260 . //construct outcome variables
1261 . gen logRegPerCapita = logregistrations-log(metropopulation + nonmetropopula
     > tion)
     (510 missing values generated)

1262 . gen logGasUsePerCapita = loghighwaygasuse-log(metropopulation + nonmetropop
     > ulation)
     (459 missing values generated)

1263 .
1264 .
1265 . //Label treatment variables
1266 .
1267 . label define nosafetyind 0 "No Treatment" 1 "Treatment"

1268 . label values nosafetyind nosafetyind

1269 .
1270 .
1271 . //Generate State Dummies
1272 . egen stateGroup = group(state)

1273 .
1274 . //Drop Datapoints that are replications of past data for registrations
1275 . drop if statecode=="CO" & year==2006
     (1 observation deleted)

1276 . drop if statecode=="IN" & (year==2006 | year==2007 | year==2009)
     (3 observations deleted)
```

```
1277 . drop if statecode=="MT" & year==2005
     (1 observation deleted)

1278 . drop if statecode=="NJ" & year==2008
     (1 observation deleted)

1279 . drop if statecode=="PR" & (year==2002 | year==2003 | year==2005 | year==200
     > 8 | year==2009 | year==2010)
     (0 observations deleted)

1280 . drop if statecode=="TX" & year==2009
     (1 observation deleted)

1281 . drop if statecode=="IL" & year==2011
     (1 observation deleted)

1282 . drop if statecode=="NH" & year==2012
     (1 observation deleted)

1283 . drop if statecode=="NY" & year==2012
     (1 observation deleted)

1284 .
1285 . //Drop Datapoints that are replications of past data for gas use
1286 . drop if statecode=="RI" & (year==2003 | year==2005)
     (2 observations deleted)

1287 .
1288 .
1289 . //Drop if data is prior to 1970
1290 . drop if year<1970
     (510 observations deleted)

1291 .
```

```
1292 . //Normalize differenece of outcome variables for identification of extreme
     > points
1293 .
1294 . sort stateGroup year

1295 . xtset stateGroup year

     Panel variable: stateGroup (unbalanced)
      Time variable: year, 1970 to 2018, but with gaps
              Delta: 1 unit

1296 .
1297 . gen DiffLogReg = d.logregistrations
     (113 missing values generated)

1298 . gen DiffLogHighwayGasUse = d.loghighwaygasuse
     (62 missing values generated)

1299 . gen DiffLogRegPerCapita = d.logRegPerCapita
     (113 missing values generated)

1300 . gen DiffLogGasUsePerCapita = d.logGasUsePerCapita
     (62 missing values generated)

1301 .
1302 . local normVarList = "DiffLogReg DiffLogHighwayGasUse DiffLogRegPerCapita Di
     > ffLogGasUsePerCapita"

1303 .
1304 . sort stateGroup

1305 .
1306 . foreach varToNorm in `normVarList'{
       2.
```

```
1307 .         by stateGroup: egen tempMean = mean(`varToNorm')
       3.         by stateGroup: egen tempSD = sd(`varToNorm')
       4.         by stateGroup: egen tempNumData = count(`varToNorm')
       5.         gen tempSE = tempSD/sqrt(tempNumData)
       6.
1308 .         gen norm`varToNorm' = (`varToNorm'-tempMean)/tempSD
       7.
1309 .         drop tempMean tempSD tempNumData tempSE
       8.
1310 .         di "`varToNorm'"
       9.
1311 . }
     (113 missing values generated)
     DiffLogReg
     (62 missing values generated)
     DiffLogHighwayGasUse
     (113 missing values generated)
     DiffLogRegPerCapita
     (62 missing values generated)
     DiffLogGasUsePerCapita

1312 .
1313 . //Generate Colorado Dummy
1314 . gen coDum1 = 0

1315 . gen coDum2 = 0

1316 . replace coDum1 = 1 if statecode=="CO" & year>=2002 & year<=2009
     (7 real changes made)

1317 . replace coDum2 = 1 if statecode=="CO" & year>=2010
     (9 real changes made)

1318 .
```

```
1319 . drop if coDum1 == 1
     (7 observations deleted)

1320 .
1321 .
1322 . //Generate dummy variable for data footnotes
1323 . gen transactionDataDummy = 0

1324 .
1325 . replace transactionDataDummy = 1 if statecode=="AR" & year>=2011
     (8 real changes made)

1326 . replace transactionDataDummy = 2 if statecode=="GA" & year>=2011
     (8 real changes made)

1327 . replace transactionDataDummy = 3 if statecode=="IA" & year>=2011
     (8 real changes made)

1328 . replace transactionDataDummy = 4 if statecode=="IL" & year>=2012
     (7 real changes made)

1329 . replace transactionDataDummy = 5 if statecode=="KY" & year>=2011
     (8 real changes made)

1330 . replace transactionDataDummy = 6 if statecode=="LA" & year>=2012
     (7 real changes made)

1331 . replace transactionDataDummy = 7 if statecode=="ME" & year>=2011 //Maine is
     >  a state with what looks to not have a consistent change
     (8 real changes made)

1332 . replace transactionDataDummy = 8 if statecode=="ME" & year>2014
     (4 real changes made)
```

```
1333 . replace transactionDataDummy = 9 if statecode=="MI" & year>=2011 //Michigan
     >  is a state with what looks to not have a consistent change
     (8 real changes made)

1334 . replace transactionDataDummy = 10 if statecode=="MI" & year>2012
     (6 real changes made)

1335 . replace transactionDataDummy = 11 if statecode=="MN" & year>=2011
     (8 real changes made)

1336 . replace transactionDataDummy = 12 if statecode=="NV" & year>=2011
     (8 real changes made)

1337 . replace transactionDataDummy = 13 if statecode=="OK" & year>=2011
     (8 real changes made)

1338 . replace transactionDataDummy = 14 if statecode=="SD" & year>=2011
     (8 real changes made)

1339 . replace transactionDataDummy = 15 if statecode=="TN" & year>=2011
     (8 real changes made)

1340 . replace transactionDataDummy = 16 if statecode=="TX" & year>=2011
     (8 real changes made)

1341 . replace transactionDataDummy = 17 if statecode=="WA" & year>=2011
     (8 real changes made)

1342 . replace transactionDataDummy = 18 if statecode=="WI" & year>=2011
     (8 real changes made)

1343 . replace transactionDataDummy = 19 if statecode=="WY" & year>=2011
     (8 real changes made)
```

```
. 
. 
. 
. //Local Data Source Controls
. 
. sort state year

. 
. python
```

```
──────────────────────────────────────────────────── python (type end to exit) ──
>>> 
>>> import numpy as np
>>> import pandas as p
>>> from sfi import Data
>>> 
>>> 
>>> def getDSCDummies(data, col, cutoff):
...     
...         tempExtState = None
...     
...         tempIndex = 0
...         stateCol = 0
...     
...         rows, cols = data.shape
...     
...         dummyExt = []
...     
...         for i in range(rows):
...                 tempDiff = float(data[i,col])
...                 tempState = data[i,stateCol]
...     
...     
...                 if(abs(tempDiff)>cutoff):
...                         tempExtState = tempState
...                         tempIndex+=1
...     
...                 if tempState==tempExtState:
...                         dummyExt.append(tempIndex)
...                 else:
...                         dummyExt.append(0)
...     
...     
...         dummyExt = np.array(dummyExt).reshape((-1,1))
...         return dummyExt
...     
```

```
>>> dataCols = ['state', 'year', 'normDiffLogReg', 'normDiffLogHighwayGasUse'
> , 'normDiffLogRegPerCapita', 'normDiffLogGasUsePerCapita']
>>>
>>> data = np.array(Data.get(dataCols, missingval=np.nan))
>>>
>>>
>>> regCutoffLow = 2.93 #Spec. 4
>>> regCutoffPrimary = 4.02 #Spec. 3
>>> regCutoffHigh = 7 #Spec. 2
>>>
>>> gasUseCutoffLow = 2.93 #Spec. 4
>>> gasUseCutoffPrimary = 4.02 #Spec. 3
>>> gasUseCutoffHigh = 7 #Spec. 2
>>>
>>> regCol = dataCols.index('normDiffLogReg')
>>> gasUseCol = dataCols.index('normDiffLogHighwayGasUse')
>>> regPerCapitaCol = dataCols.index('normDiffLogRegPerCapita')
>>> gasUsePerCapitaCol = dataCols.index('normDiffLogGasUsePerCapita')
>>>
>>> rows, cols = data.shape
>>>
>>> regDummyExtLow = getDSCDummies(data, regCol, regCutoffLow)
>>> regDummyExtPrimary = getDSCDummies(data, regCol, regCutoffPrimary)
>>> regDummyExtHigh = getDSCDummies(data, regCol, regCutoffHigh)
>>>
>>> gasUseDummyExtLow = getDSCDummies(data, gasUseCol, gasUseCutoffLow)
>>> gasUseDummyExtPrimary = getDSCDummies(data, gasUseCol, gasUseCutoffPrimar
> y)
>>> gasUseDummyExtHigh = getDSCDummies(data, gasUseCol, gasUseCutoffHigh)
>>>
>>> regPerCapitaDummyExtLow = getDSCDummies(data, regPerCapitaCol, regCutoffL
> ow)
>>> regPerCapitaDummyExtPrimary = getDSCDummies(data, regPerCapitaCol, regCut
> offPrimary)
>>> regPerCapitaDummyExtHigh = getDSCDummies(data, regPerCapitaCol, regCutoff
> High)
>>>
>>> gasUsePerCapitaDummyExtLow = getDSCDummies(data, gasUsePerCapitaCol, gasU
> seCutoffLow)
>>> gasUsePerCapitaDummyExtPrimary = getDSCDummies(data, gasUsePerCapitaCol,
> gasUseCutoffPrimary)
>>> gasUsePerCapitaDummyExtHigh = getDSCDummies(data, gasUsePerCapitaCol, gas
> UseCutoffHigh)
>>>
>>> Data.addVarInt('regDummyExtSpec4')
>>> Data.store("regDummyExtSpec4",None, regDummyExtLow[:])
```

```
>>> Data.addVarInt('regDummyExtSpec3')
>>> Data.store("regDummyExtSpec3",None, regDummyExtPrimary[:])
>>> Data.addVarInt('regDummyExtSpec2')
>>> Data.store("regDummyExtSpec2",None, regDummyExtHigh[:])
>>>
>>> Data.addVarInt('gasUseDummyExtSpec4')
>>> Data.store("gasUseDummyExtSpec4",None, gasUseDummyExtLow[:])
>>> Data.addVarInt('gasUseDummyExtSpec3')
>>> Data.store("gasUseDummyExtSpec3",None, gasUseDummyExtPrimary[:])
>>> Data.addVarInt('gasUseDummyExtSpec2')
>>> Data.store("gasUseDummyExtSpec2",None, gasUseDummyExtHigh[:])
>>>
>>> Data.addVarInt('regPerCapitaDummyExtSpec4')
>>> Data.store("regPerCapitaDummyExtSpec4",None, regPerCapitaDummyExtLow[:])
>>> Data.addVarInt('regPerCapitaDummyExtSpec3')
>>> Data.store("regPerCapitaDummyExtSpec3",None, regPerCapitaDummyExtPrimary[
> :])
>>> Data.addVarInt('regPerCapitaDummyExtSpec2')
>>> Data.store("regPerCapitaDummyExtSpec2",None, regPerCapitaDummyExtHigh[:])
>>>
>>> Data.addVarInt('gasPerCapitaDummyExtSpec4')
>>> Data.store("gasPerCapitaDummyExtSpec4",None, gasUsePerCapitaDummyExtLow[:
> ])
>>> Data.addVarInt('gasPerCapitaDummyExtSpec3')
>>> Data.store("gasPerCapitaDummyExtSpec3",None, gasUsePerCapitaDummyExtPrima
> ry[:])
>>> Data.addVarInt('gasPerCapitaDummyExtSpec2')
>>> Data.store("gasPerCapitaDummyExtSpec2",None, gasUsePerCapitaDummyExtHigh[
> :])
>>>
>>>
>>> end
```

```
. 
. 
. //Generate normalized year values to reduce differences in magnitude of cov
> ariates.
. //Scale year for better numerical performance
. local yearScale = 1

. gen normYear = year/`yearScale'

. gen quadYear = (year/1)^2

. 
. //Establish treatment interacted with pre-treatment population share in met
> ro areas
. frame copy default MetroShares

. frame change MetroShares

. keep if year==1970
(2,429 observations deleted)

. 
. cap gen preTreatMetroShare = metropopulation/(metropopulation+nonmetropopul
> ation)

. 
. frame change default

. 
. cap frlink m:1 state, frame(MetroShares)

. cap frget preTreatMetroShare, from(MetroShares)

. 
```

```
1373 . cap gen metroIntTreatment = nosafetyind*preTreatMetroShare

1374 .
1375 . frame drop MetroShares

1376 .
     end of do-file

1377 . do GasUseIVEmissions

1378 . eststo clear

1379 .
1380 . if("`c(os)'"=="Windows"){
1381 .         local tableSaveDir = "ReviewResponseResults\"
1382 .         local plotSaveDir = "ReviewResponseResults\"
1383 .         local pythonSaveDir = "PythonScripts\"
1384 . }

1385 . else{
1386 .         local tableSaveDir = "ReviewResponseResults/"
1387 .         local plotSaveDir = "ReviewResponseResults/"
1388 .         local pythonSaveDir = "PythonScripts/"
1389 . }

1390 .
1391 . //Establish Control Variable Sets
1392 . //Spec1 controls are the same for both registrations and fuel use
1393 . local spec1Controls = "c.normYear#i.stateGroup c.quadYear#i.stateGroup asin
     > hMetroPop asinhNonMetroPop asinhMetroInc asinhNonMetroInc logrealmeangaspri
     > ce logemployment loglicenseddrivers logrealstategdp logroadmileage"

1394 .
1395 . //Other controls are not the same for both registrations and VMT
```

```
1396 . //Registration Controls
1397 . local spec2RegControls = "`spec1Controls' i.transactionDataDummy"

1398 . local spec3RegControls = "`spec2RegControls' i.regDummyExtSpec3"

1399 . local spec4RegControls = "`spec2RegControls' i.regDummyExtSpec4"

1400 .
1401 .
1402 . di "`spec3RegControls'"
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.transactionDataDummy i.regDummy
     > ExtSpec3

1403 .
1404 . //Gas use Controls
1405 . local spec2GasUseControls = "`spec1Controls'"

1406 . local spec3GasUseControls = "`spec2GasUseControls' i.gasUseDummyExtSpec3"

1407 . local spec4GasUseControls = "`spec2GasUseControls' i.gasUseDummyExtSpec4"

1408 .
1409 . //Absorb Vars are always time and state fixed effects
1410 . local absorbVars = "ib51.stateGroup ib1995.year"

1411 .
1412 . //Establish Cluster Variable
1413 . local clusterVar = "stateGroup"

1414 .
1415 . //Establish primary null
1416 . local primaryNull = 11
```

```
1417 . local altNull = round(9.7, .1)

1418 . local altNull2 = 1

1419 . local altNull3 = -4.8

1420 . local altNull4 = -6.1

1421 .
1422 . frame create emissions

1423 . frame change emissions

1424 .
1425 . if("`c(os)'"=="Windows"){
1426 .         local dataFile = "CleanedData\EmissionsProgramsV2.xlsx"
1427 . }

1428 . else{
1429 .         local dataFile = "CleanedData/EmissionsProgramsV2.xlsx"
1430 . }

1431 .
1432 . import excel using "`dataFile'", first sheet("Clean") case(l)
    (4 vars, 51 obs)

1433 .
1434 . frame change default

1435 . frlink m:1 state, frame(emissions)
      (all observations in frame default matched)

1436 . frget stateemissionsprogram startemission endemission, from(emissions)
    (734 missing values generated)
    (734 missing values generated)
      (3 variables copied from linked frame)
```

```
1437 .
1438 . gen emissionind = 0

1439 . replace emissionind = 1 if year>=startemission & year<=endemission & statee
     > missionsprogram==1
     (1,092 real changes made)

1440 .
1441 . frame drop emissions

1442 .
1443 . //non-normalized models
1444 . local treatmentVar = "1.nosafetyind"

1445 .
1446 . local fsVarsSpec2 = "logregistrations i.nosafetyind `spec2RegControls' i.em
     > issionind"

1447 . local rfVarsSpec2 = "loghighwaygasuse i.nosafetyind `spec2GasUseControls' i
     > .emissionind"

1448 .
1449 . local fsOut = "logregistrations"

1450 . local rfOut = "loghighwaygasuse"

1451 .
1452 . local treatControl = "nosafetyind"

1453 .
1454 . local fsControlsSpec1 = "`spec1Controls' i.emissionind"

1455 . local rfControlsSpec1 = "`spec1Controls' i.emissionind"
```

```
1456 .
1457 . local modelTitleSpec1 = "Spec1GasUseEmissions"

1458 . local nullLowSpec1 = -4

1459 . local nullDeltaSpec1 = 0.001

1460 . local nullHighSpec1 = 1

1461 .
1462 . runModelFWL `fsOut', `rfOut', `fsControlsSpec1', `rfControlsSpec1', `treatC
     > ontrol', `absorbVars', `clusterVar', `treatmentVar', `modelTitleSpec1', `nu
     > llLowSpec1', `nullDeltaSpec1', `nullHighSpec1', `primaryNull', `altNull', `
     > altNull2', `altNull3', `altNull4'
     logregistrations
     loghighwaygasuse
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.emissionind
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.emissionind
     nosafetyind
     ib51.stateGroup ib1995.year
     Working on:Spec1GasUseEmissions

          Source │       SS           df       MS      Number of obs   =     2,42
     > 9
     ────────────┼──────────────────────────────────   F(1, 2428)      =     33.0
     > 6
           Model │  .083307463         1  .083307463   Prob > F        =    0.000
     > 0
        Residual │  6.11761338     2,428   .00251961   R-squared       =    0.013
     > 4
     ────────────┼──────────────────────────────────   Adj R-squared   =    0.013
     > 0
           Total │  6.20092085     2,429   .00255287   Root MSE        =     .050
     > 2
```

```
------------------------------------------------------------------------------
> -------
logregistrations~s |  Coefficient  Std. err.      t    P>|t|     [95% conf. in
> terval]
-------------------+----------------------------------------------------------
> -------
fs_nosafetyind_res |   .0533329   .0092751     5.75   0.000     .0351449    .
> 0715209
------------------------------------------------------------------------------
> -------

scalars:
              r(covar) =  -.0000309513218122
           r(covarLow) =  -.0000309513218122
          r(covarHigh) =  -.0000309513218122
           r(deltaVar) =  .000096423943749
              r(piVar) =  .0002077284376947
   r(covarUncorrected) =  -.0000283697954072
  r(deltaVarUncorrect
    ed)               =  .0000883816068701
   r(piVarUncorrected) =  .0001904026365471
              r(delta) =  -.0148651739251812
                 r(pi) =  .053332921149737
        r(rfCorrection) =  1.090995594713656
        r(fsCorrection) =  1.090995594713656
                  r(N) =  2429
             r(nClust) =  51
                r(rfK) =  159
                r(fsK) =  159

1463 .
1464 . local fsControlsSpec2 = "`spec2RegControls' i.emissionind"

1465 . local rfControlsSpec2 = "`spec2GasUseControls' i.emissionind"
```

```
1466 .
1467 . local modelTitleSpec2 = "Spec2GasUseEmissions"

1468 . local nullLowSpec2 = -4

1469 . local nullDeltaSpec2 = 0.001

1470 . local nullHighSpec2 = 1

1471 .
1472 . runModelFWL `fsOut', `rfOut', `fsControlsSpec2', `rfControlsSpec2', `treatC
     > ontrol', `absorbVars', `clusterVar', `treatmentVar', `modelTitleSpec2', `nu
     > llLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primaryNull', `altNull', `
     > altNull2', `altNull3', `altNull4'
     logregistrations
     loghighwaygasuse
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.transactionDataDummy i.emission
     > ind
     c.normYear#i.stateGroup c.quadYear#i.stateGroup asinhMetroPop asinhNonMetroPo
     > p asinhMetroInc asinhNonMetroInc logrealmeangasprice logemployment loglicen
     > seddrivers logrealstategdp logroadmileage i.emissionind
     nosafetyind
     ib51.stateGroup ib1995.year
     Working on:Spec2GasUseEmissions

          Source │       SS           df       MS      Number of obs   =     2,42
     > 9
     ────────────┼──────────────────────────────────   F(1, 2428)      =     35.2
     > 2
           Model │  .080349927         1  .080349927   Prob > F        =    0.000
     > 0
        Residual │  5.53875665     2,428  .002281201   R-squared       =    0.014
     > 3
     ────────────┼──────────────────────────────────   Adj R-squared   =    0.013
     > 9
           Total │  5.61910658     2,429  .002313342   Root MSE        =    .0477
     > 6
```

```
──────────────────────┬──────────────────────────────────────────────────────────
> ──────
logregistrations~s │ Coefficient  Std. err.      t    P>|t|     [95% conf. in
> terval]
──────────────────────┼──────────────────────────────────────────────────────────
> ──────
fs_nosafetyind_res │    .054291   .0091478     5.93   0.000     .0363527    .
> 0722294
──────────────────────┴──────────────────────────────────────────────────────────
> ──────

scalars:
              r(covar) =  -.0000332510662156
           r(covarLow) =  -.0000332510662156
          r(covarHigh) =  -.0000329727533266
           r(deltaVar) =  .0000972378286585
              r(piVar) =  .0001956769974189
   r(covarUncorrected) =  -.0000302226273748
  r(deltaVarUncorrect
    ed)               =  .0000883816068701
   r(piVarUncorrected) =  .0001778551382522
              r(delta) =  -.0148651739251812
                 r(pi) =  .0542910360724894
        r(rfCorrection) =  1.090995594713656
        r(fsCorrection) =  1.100204353620613
                  r(N) =  2429
             r(nClust) =  51
                r(rfK) =  159
                r(fsK) =  178

1473 . // runModel `fsVarsSpec2', `rfVarsSpec2', `clusterVar', `treatmentVar', `mo
     > delTitleSpec2', `nullLowSpec2', `nullDeltaSpec2', `nullHighSpec2', `primary
     > Null', `altNull', `altNull2', `altNull3', `altNull4'
1474 . //
```

```
1475 . //
1476 . // // //Combine all of the IV information
1477 . // // python script "`pythonSaveDir'combineIV.py"
1478 . // //
1479 . // // //Output tables
1480 . // // constructTable GasUsePerCapitaiv; Fleet Gas Use Per Capita; Spec1GasU
     > sePerCapita Spec2GasUsePerCapita Spec3GasUsePerCapita Spec4GasUsePerCapita;
     >  `primaryNull'; `altNull'; `altNull2'; `altNull3'; `altNull4'
1481 . // // constructTable GasUseiv; Fleet Gas Use; Spec1GasUse Spec2GasUse Spec3
     > GasUse Spec4GasUse; `primaryNull'; `altNull'; `altNull2'; `altNull3'; `altN
     > ull4'
1482 .
     end of do-file

1483 .
1484 .
1485 . //End Timer
1486 . qui timer off 1

1487 . qui timer list 1

1488 . local minToComplete = `r(t1)'/60

1489 . if(`r(t1)'<60){
1490 .         di "Script took `r(t1)' seconds to complete."
1491 . }

1492 . else{
1493 .         di "Script took `minToComplete' minutes to complete."
     Script took 1.236283333333333 minutes to complete.
1494 . }

1495 .
1496 . cd "ReviewResponseResults"
     /Users/connorforsythe/Library/CloudStorage/Box-Box/CMU/Marginal Mileage Proje
     > ct/Publication Replication Code/JAERE/PrimaryModel/ReviewResponseResults

1497 .
```