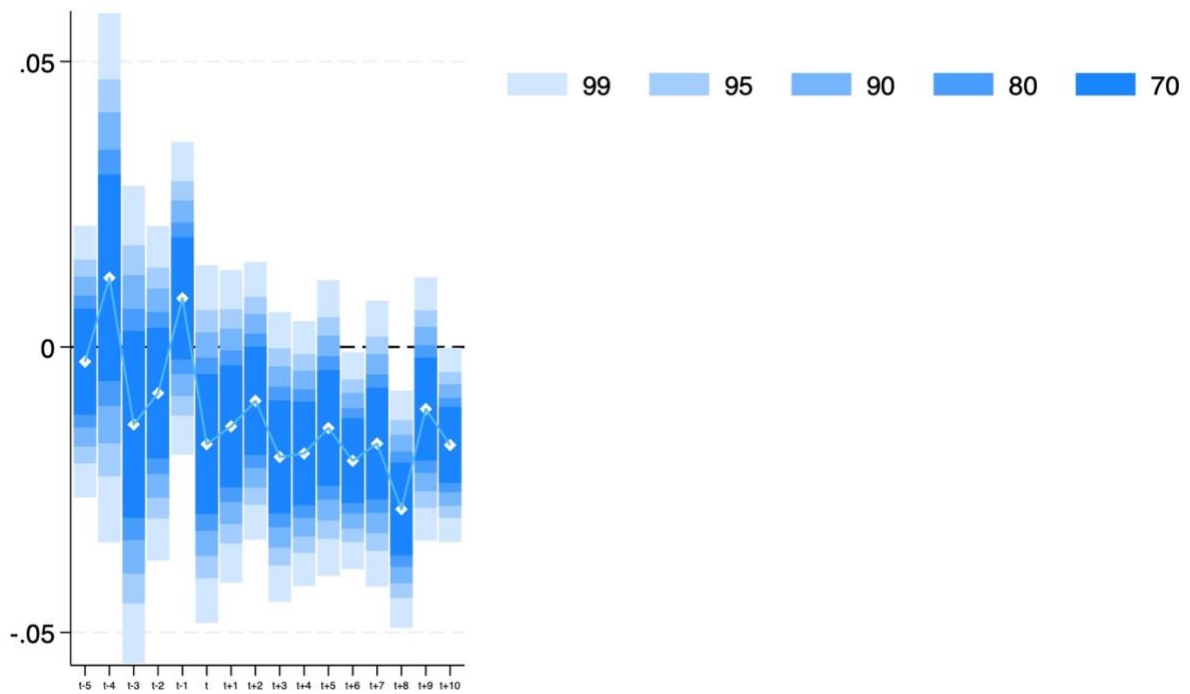
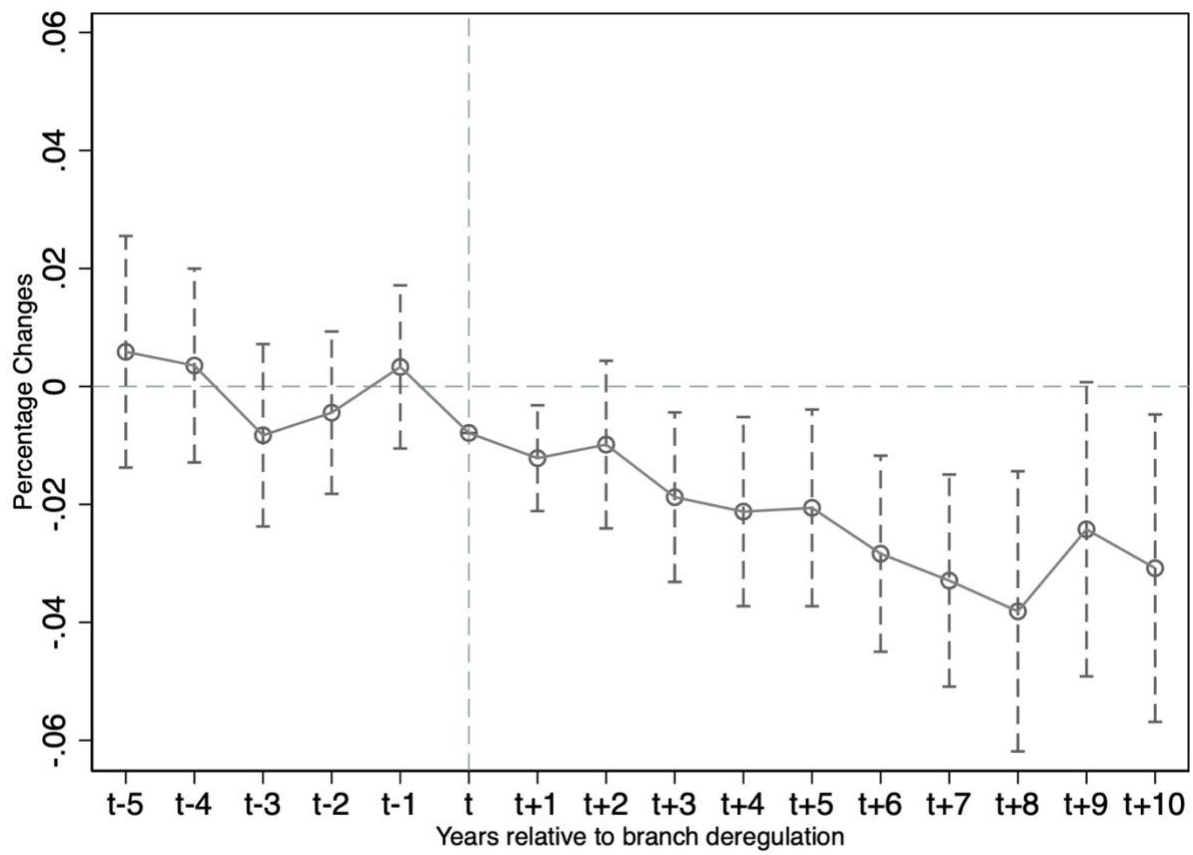


Figure 1. Pre- and Post-bank deregulation Estimation of the Average Treatment Effect



Note: tvdiff method

Figure 2. Pre- and Post-bank deregulation Estimation of the Average Treatment Effect



Note: coefplot method

```
set more off
clear all
```

```
*
```

```
*
```

-3.1- DID

```
*** Read in the raw data ***
```

```
use "/Users/qilinzhou/Desktop/Stata-econometric/StataDemo8/lec3_macro.dta",
clear
```

**** [Description]**

```
**      Beck et al. (2010) is a classic paper using a multi-period DID model published
in the Journal of Finance.
```

```
**      The paper examines the impact of bank branch deregulation on income
distribution inequality in the U.S. states
```

```
**      that deregulated bank branches at various points during the 1960-1999
period.
```

```
**      The sample consists of 49 U.S. states and 31 years (1976-2006) of balanced
panel data, with a total of 1519 observations.
```

```
**      The paper selects four types of indicators, including the Gini coefficient,
**      as proxies for the inequality of the dependent variable, and only the Gini
coefficient is selected
```

```
**      to demonstrate the treatment effect of time-varying DID with inconsistent
policy points in time.
```

```
**      The treatment time dummy variable takes a value of 1 after bank branch
deregulation in a state,
```

```
**      implying that the state is in the treatment group thereafter.
```

```
**      This time-varying DID model is set up as a two-way fixed effects model,
**      so individual fixed effects and time fixed effects are controlled for in the
model
```

```
**      to produce area dummy variables and time dummy variables, respectively.
```

```
label var _intra "Bank deregulation" //treatment dummy variable
```

```
xtset statefip wrkyr //declare the panel data setting
```

```

    tabulate wrkyr, gen(wrkyr_dumm) //generate dummies of time fixed effects
    tabulate statefip, gen(state_dumm) //generate dummies of individual fixed effects

    replace p10 = 1 if p10==0

    generate log_gini = log(gini) //as y

*without control variables
    xtreg log_gini _intra wrkyr_dumm*, fe robust

*define the marco of controls
    global Xs "gsp_pc_growth prop_blacks prop_dropouts prop_female_headed
unemploymentrate"

*with control variables
    xtreg log_gini _intra $Xs wrkyr_dumm*, fe robust
    des

*Graph for Parallel-trend tests

    //Useful user-written commands for DID
    ssc install coefplot,replace
    ssc install tvdiff,replace

    //The first approach: *tvdiff*
    generate D = (wrkyr - branch_reform == 0)
    generate y = ln(gini)

    global X "gsp_pc_growth prop_blacks prop_dropouts"

    tvdiff y D $X, model(fe) pre(5) post(10) vce(robust) test_tt graph
save_graph(mygraph)

    //The second approach: *coefplot*
    gen policy = wrkyr - branch_reform
    replace policy = -5 if policy <= -5
    replace policy = 10 if policy >= 10

    gen policy_d = policy + 5

    xtreg y ib5.policy_d i.wrkyr, fe r

    ///generate the average values of the first 5 periods
    forvalues i = 0/4{

```

```

        gen b_`i' = _b[`i'.policy_d]
    }

```

```

gen avg_coef = (b_0+b_4+b_3+b_2+b_1)/5
sum avg_coef

```

```

coefplot, baselevels ///
    drop(*.wrkyr_cons policy_d) ///
    coeflabels(0.policy_d = "t-5" ///
1.policy_d = "t-4" ///
2.policy_d = "t-3" ///
3.policy_d = "t-2" ///
4.policy_d = "t-1" ///
5.policy_d = "t" ///
6.policy_d = "t+1" ///
7.policy_d = "t+2" ///
8.policy_d = "t+3" ///
9.policy_d = "t+4" ///
10.policy_d = "t+5" ///
11.policy_d = "t+6" ///
12.policy_d = "t+7" ///
13.policy_d = "t+8" ///
14.policy_d = "t+9" ///
15.policy_d = "t+10") ///
    vertical ///
    yline(0, lwidth(vthin) lpattern(dash) lcolor(teal)) ///
    ylabel(-0.06(0.02)0.06) ///
    xline(6, lwidth(vthin) lpattern(dash) lcolor(teal)) ///
    ytitle("Percentage Changes", size(small)) ///
    xtitle("Years relative to branch deregulation", size(small)) ///
    transform(*=@-r(mean)) ///
    addplot(line @b @at) ///
    ciopts(lpattern(dash) recast(rcap) msize(medium)) ///
    msymbol(circle_hollow) ///
    scheme(s1mono)

```