```
*This do-file creates the analysis dataset
cap log close
log using "$log_save\create_analysis_datasets", replace

foreach dataset in unbalanced_survivors unbalanced balanced survivors {
                use  "$saveaddress_data\analysis_firm.dta", clear

                drop if c_type == 1
                drop if c_type == 2
                cap drop _merge
                merge 1:1 taxrefno taxyear using "Z:\Master Data\Auxillary capital sto
> ck\Imputed_capital_stock_beta.dta",  keepusing(pi_iv_fixed_pd_10) keep(master matche
> d)
                drop _merge
/*
                *FYE ADJUSTMENT
                cap drop month
                gen month = month(FYE)
                keep if month ==2
                */
        *CLEAN THE PROVINCE VARIABLE
                drop if busprov_geo == "EXCEPTION"
                cap drop busprov_geo_num
                egen busprov_geo_num = group(busprov_geo)
                bys taxrefno: egen busprov_geo_num_imp = mode(busprov_geo_num)

                *ADD THE RAINFALL VARIABLE
                cap drop province
                gen province =  busprov_geo
                cap drop _merge
                sort taxyear province

           merge m:m  taxyear using "$saveaddress_data\Annual rainfall by province in
> line with taxyears.dta",  keepusing(rainfall) keep(master matched)


        *KEEP ONLY AGRIC SUBSECTORS
                keep if imp_mic_sic7_3d ==11 | imp_mic_sic7_3d== 12 | imp_mic_sic7_3d
> ==13| imp_mic_sic7_3d== 14 | imp_mic_sic7_3d== 15




        *DROP 2008 2009  2010 and 2018
                drop if taxyear == 2008 | taxyear == 2009 | taxyear== 2010 |  taxyear
> ==2018
                tab taxyear

        *DROP DUPLICATES
                cap drop n
                bysort taxrefno taxyear : gen n= _n
                count if n ==1



                *ADJUST THE FOREIGN FIRM DUMMY
                replace ITR14_c_foreign_broad=0 if ITR14_c_foreign_broad==.

        *CREATE THE CPI VARIABLE

                cap drop cpi
                gen cpi = .
                replace cpi = (92.98/71.13) if taxyear ==2011
                replace cpi = (92.98/74.97) if taxyear ==2012
                replace cpi = (92.98/79.13) if taxyear ==2013
                replace cpi = (92.98/83.72) if taxyear ==2014
                replace cpi = (92.98/88.58) if taxyear ==2015
                replace cpi = (92.98/92.98) if taxyear ==2016
                replace cpi = (92.98/98.85) if taxyear ==2017
        *DEFLATE USING CPI

                replace g_cos = g_cos*cpi*100
                replace g_sales = g_sales*cpi*100
                replace k_ppe = k_ppe*cpi*100
                replace k_faother = k_faother*cpi*100
                replace x_labcost = x_labcost*cpi*100
                replace tot_kerr = tot_kerr*cpi*100


        *VALUE-ADDED
                cap drop value_added
                drop if g_cos == 0 | g_cos==.
                drop if g_sales == 0 | g_sales == .
                gen value_added = g_sales - g_cos
                replace value_added=x_labcost+g_grossprofit if value_added==.|value_ad
> ded<0
                replace value_added=. if value_added==.|value_added<0
                drop if value_added== 0 | value_added ==.
                sum value_added

        *EMPLOYMENT
                cap drop employment
                drop if irp5_kerr_weight_b == 0 | irp5_kerr_weight_b==.
                gen employment = irp5_kerr_weight_b
                sum employment
                label var employment "Employment"

        *FIXED CAPITAL
                cap drop capital
                gen capital = pi_iv_fixed_pd_10
                sum capital, d
                label var capital "Fixed_Capital"


        *COST OF SALE
                cap drop cost_sales
                drop if g_cos == 0 | g_cos == .
                drop if g_cos < 0
                gen cost_sales = g_cos
                sum cost_sales
                labe var cost_sales "Cost_of_sales"

        *REVENUE
                cap drop revenue
                gen revenue = g_sales
                label var revenue "Sales_revenue"

        *OPERATING PROFIT
                cap drop oprofits
                gen oprofits  = revenue - cost_sales
                sum oprofits

        *TOTAL FACTOR PRODUCTIVITY  [OLS METHOD FOR NOW]
                gen va = log(value_added)
                cap drop l
                gen l = log(employment)
                label var l "Employment"
                cap drop k
                gen k = log(capital)
                label var k "Capital"


                xi: reg va l k, cluster(n_fid)
                cap drop tfp
                gen tfp = va - (_b[l]*l + _b[k]*k)
        *GENERATE FARM SIZE VARIABLE [USING EMPLOYEMNT ]
                cap drop size_emp
                gen size_emp = .
                replace size_emp =1 if employment <20
                replace size_emp  =2 if employment >=20 & employment <50
                replace size_emp = 3 if employment >=50

        *GENERATE PRE-POLICY SIZE
                br n_fid taxyear c_type
                cap drop c_type_mode
                bys n_fid: egen c_type_mode = mode(c_type)
                gen c_type_adj = c_type_mode
                cap drop size_ctype_adj
                recode c_type_adj (3=1 "Micro") (4=2 "Small") (5=3 "Medium to large")
> , gen(size_ctype_adj)


        *GENERATE FARM SIZE VARIABLE [USING GROSS INCOME AND TOTAL ASSETS]
                *cap drop size_ctype
                gen size_ctype = c_type


        *GENERATE FARM SIZE VARIABLE [USING MARLIES SUGGESTION]


        *LABOUR COST
                sum x_labcost
                drop if x_labcost==0 | x_labcost== .
                gen labour_cost = x_labcost
                label var labour_cost "Labour_cost"

        *FRACTION OF AFFECTED WORKERS
                sum fa_use, d
                label var fa_use "Fraction_Affected"

        *HHI INDEX
                cap drop tot_ind_sales
                bys imp_mic_sic7_4d: egen tot_ind_sales = total(g_sales)
                cap drop share
                gen share = (g_sales/tot_ind_sales)*100
                cap drop share_sq
                gen share_sq = share^2
                cap drop hhi
                bys imp_mic_sic7_4d: egen hhi = total(share_sq)
                sum hhi, d


        *VALUE-ADDED PER WORKER
                cap drop value_added_pe
                gen value_added_pe  = value_added/employment

        *CAPITAL-LABOUR RATIO
                cap drop cap_lab
                gen cap_lab = capital/employment

        *REVENUE PER EMPLOYEE
                cap drop rev_pe
                gen rev_pe = revenue/employment

        *OPERATING PROFIT PER EMPLOYEE
                cap drop oprofits_pe
                gen oprofits_pe = oprofits/employment

        *VALUE-ADDED PER CAPITAL
                cap drop value_added_cap
                gen value_added_cap = value_added/ capital
        *LABOUR COST PER EMPLOYEE
                sum x_labcost, d
                replace x_labcost = r(p1) if x_labcost < r(p1)
                cap drop labcost_pe
                gen labcost_pe = x_labcost/employment

                cap drop llabcost_pe
                gen llabcost_pe = log(labcost_pe)

                preserve
                local keep_level = "employment capital revenue labour_cost cost_sales
> fa_use"
                keep taxrefno taxyear  size_ctype_adj  `keep_level'
                save "$saveaddress_data\sumstat_data", replace
                restore

        *WINSORIZE AT 1%

                local keep_level = "employment capital revenue labour_cost cost_sales
> fa_use"
                levelsof taxyear, loca(levels)
                foreach level of local levels{
                foreach var of local keep_level{
             sum `var' if taxyear == `level', d

                 replace `var' = r(p1) if `var'< r(p1) & taxyear == `level '
                 replace `var' = r(p99) if `var' > r(p99) & taxyear== `level'


         }
 }



        *LOG | VALUE-ADDED PER WORKER
                cap drop lvalue_added_pe
                gen lvalue_added_pe  = log(value_added_pe )

        *LOG | CAPITAL-LABOUR RATIO
                cap drop lcap_lab
                gen lcap_lab = log(cap_lab)
                label var lcap_lab "Capital_intensity"
        *LOG | REVENUE
                cap drop lrevenue
                gen lrevenue = log(revenue)
                label var lrevenue "Revenue"

        *LOG | REVENUE - PER WORKER
                cap drop lrev_pe
                gen lrev_pe = log(rev_pe)
                label var  lrev_pe "Revenue per worker"


        *LOG | TOTAL FACTOR PRODUCTIVITY
                cap drop ltfp
                gen ltfp = tfp
                label var ltfp "Total_Factor_Productivity"

        *LOG | OPERATING PROFIT PER WORKER
                cap drop loprofits_pe
                gen loprofits_pe = log(oprofits_pe)
                label var loprofits_pe "Operating_profit_per_worker"

        *LOG | VALUE ADDED PER CAPITAL

                gen lvalue_added_cap = log(value_added_cap)

        *LOG | LABOUR COST
                gen l_labcost = log(x_labcost)
                label var l_labcost "Total_labour_cost"
```

```
*LOG | TOTAL WAGES {AMOUNT 3601}
        gen l_wages = log(tot_3601)
        label var l_wages "Total_wages"

*LOG | TOTAL COST OF SALES
        cap drop lcogs
        gen lcogs = log(g_cos)

*LOG | MATERIALS

        cap drop materials
        gen materials = g_cos - tot_kerr
        gen lmaterials = log(materials)
        label var lmaterials "Material_cost"

*LOG | SALES
        cap drop lsales
        gen lsales = log(g_sales)


*LOG | OPERATING PROFITS
        gen loprofit = log(oprofits )
        label var loprofit "Operating_profit"


*LOG | LABOUR COST PER CAPITAL

        gen labcost_capcost = x_labcost/capital
        gen llabcost_capcost=log(labcost_capcost)
        label var llabcost_capcost "Labour_cost_per_capital"

*LOG | LABOUR COST PER EMPLOYEE

        cap drop llabcost_pe
        gen llabcost_pe  = log(labcost_pe)
        label var llabcost_pe "Labour_cost_per_worker"

*LOG | AVERAGE WAGE

        cap drop  awage
        gen awage = tot_3601/ employment
        gen lawage = log(awage)
        label var lawage "Average_wage"


*LOG | NON-WAGE LABOUR COSTS

        gen non_wage_labcosts = x_labcost - tot_kerr
        gen lnon_wage_labcosts = log(non_wage_labcosts)
        label var lnon_wage_labcosts "Non-wage_labour_cost"

*LOG | EMPLOYEE TRAINING EXPENDITURE
        cap drop ltrain
        gen ltrain = log(ITR14_x_training)

        label var ltrain "Employee_training_expenditure"

*EXIT
sort taxrefno taxyear

cap drop n
cap drop N
bys n_fid: gen n = _n
bys n_fid: gen N = _N
tab taxyear
br taxrefno n_fid taxyear n N

xtset FID taxyear
```

```
        gen exit =0
        replace exit = 1 if N==n & n!=.
        labe var exit "Firm_exit"

        replace exit = 0 if taxyear <2013

         cap drop n
        cap drop N
        local keep_log = " ltrain lnon_wage_labcosts lawage llabcost_pe llabcost_capco
> st loprofit lmaterials l_labcost  loprofits_pe ltfp lcap_lab lrevenue lrev_pe exit "


   *add wages
   cap drop _merge
   merge 1:1 taxrefno taxyear using  "Z:\Workbenches\epadmin\michael_kilumelume\2024 p
> rojects\minimum wage\data\seasonal_nonseasonal_lowhigh.dta", keep(master matched) ke
> epusing(ljobs_total ljobs_seasonal ljobs_nonseasonal ljobs_low_wage ljobs_high_wage
> lavg_seasonal_wage lavg_nonseasonal_wage lavg_low_wage lavg_high_wage min_low)


  *Restrict dataset to farms with some worker consistently earning lower then R5400
  drop if min_low ==0



********************************************************************************
*define treatment and period
********************************************************************************
        *define treatment group based on fractio affected
                cap drop t
                gen t=0
                replace t = 1 if fa> 0

        *Define pre and post period
                cap drop p
                gen p = 0
                replace p = 1 if taxyear >=2014

        *create constant controls
                sort n_fid taxyear

foreach var in   tfp  cap_lab  rainfall employment revenue k tot_3601  {
                cap drop tmp_`var'
                bys n_fid: egen tmp_`var' = mean(`var') if p==0

                cap drop mn_`var'
                bys n_fid: egen mn_`var' = min(tmp_`var')

                cap drop lmn_var
                gen lmn_`var' = log(mn_`var')

                cap drop tmp_*

}
*DROP FIRMS ENTERING AFTER THE POLICY

                sort n_fid taxyear
                br n_fid taxyear
                cap drop n
                bys n_fid: gen n= _n

                gen entry_after_tg = 0
                replace entry_after_tg = 1 if n==1 & taxyear > 2013

                bys n_fid: egen drop_afer = max(entry_after_tg)
                drop if drop_afer== 1

                cap drop n
```

```
if "`dataset'"== "balanced"{
        *create rectangular dataset for a balanced panel
                drop if taxrefno ==""
                egen id_new_num = group(id_new)
                xtset n_fid taxyear
                tsfill , full



foreach var of local keep_log {
                replace `var' = 0 if `var'==. & taxyear >2013
}
                replace fa_use =fa_use[_n-1] if fa_use==0 & taxyear > 2013 & fa_use[_n
> -1]! = 0 & FID==FID[_n-1]

}

if "`dataset'" == "survivors" {
        *Tag survivors
                br FID taxyear
                bys FID : gen N = _N

                *Identify farms entering in 2012
                cap drop n
                bys FID : gen n =_n

                gen enter_2012_tag = 1 if n ==1 & taxyear == 2012
                bys taxrefno: egen enter_2012 = max(enter_2012_tag)

                gen survivor = 1 if N ==7
                replace survivor = 1 if N==6 & enter_2012 ==1

                keep if survivor ==1
}

if "`dataset'" == "unbalanced"{
    *Tag survivors
                br FID taxyear
                bys FID : gen N = _N

                *Identify farms entering in 2012
                cap drop n
                bys FID : gen n =_n

                gen enter_2012_tag = 1 if n ==1 & taxyear == 2012
                bys taxrefno: egen enter_2012 = max(enter_2012_tag)

                gen survivor = 1 if N ==7
                replace survivor = 1 if N==6 & enter_2012 ==1

        drop if survivor ==1

}


                save "$saveaddress_data\\analysis_`dataset'.dta", replace

}

log close
```