

STATA and SAS Code for “Consequences for Culpable Auditors”

Our code is split into two parts. Part 1 displays code for analyses of the culpable auditor sample. Part 2 presents code for analyses of the comparison auditor sample.

Part 1

Code for Analyses on Culpable Auditor Sample

We manually collect raw data on culpable auditors from the various sources identified in the data description file and organize it into an Excel spreadsheet. The following STATA code is used to generate descriptive statistics and conduct multivariate analyses for labor market consequences for culpable auditors.

```
*****  
*sample selection table 1 Panel A  
*****  
import delimited "C:\Users\mengl\OneDrive - Temple University\Desktop\Temple\Audit Partner  
AAER\JAR\Data\aaer resp.csv", clear //obtained directly from audit analytics aaer database  
gen date_new=date(date, "YMD")  
gen year=year(date_new)  
keep if year>2002&year<2020 //sample period from 2003 to 2019  
  
egen unique_aerno = tag(aaerreleaseno)  
egen unique_aaerevent = tag(aaereventkey)  
count if unique_aerno == 1 // 2,318 total number of enforcement orders against any party  
count if unique_aaerevent == 1 // 1,053 unique number of events subject to enforcement  
  
import excel "C:\Users\mengl\OneDrive - Temple University\Desktop\Temple\Audit Partner  
AAER\JAR\Data\SEC.KLMP.xlsx", sheet("AAER audit related") firstrow clear //import dataset  
with manually collected auditor aaers  
  
egen unique_aaer=tag(AAER_No) if Individual_audit==1  
count if unique_aaer==1 //270 enforcement orders name one or more individuals  
sum Individual_audit //obs 338 unique individuals at an accounting firm named in enforcement order  
sum Individual_audit if US==0 // 22 foreign individuals  
sum Individual_audit if PCAOBregistrationissue==1 //41 failed to register with PCAOB in 2007 and  
2008  
  
import excel "C:\Users\mengl\OneDrive - Temple University\Desktop\Temple\Audit Partner  
AAER\JAR\Data\PCAOB.KLMP.xlsx", sheet("PCAOB enforcement") firstrow clear //import  
dataset with manually collected PCAOB enforcement  
  
keep if Year<2020  
egen unique_case = tag(Case)  
egen unique_order=tag(PCAOB_ReleaseNo)  
count if unique_case == 1 //250 unique events  
count if unique_order == 1 //320 total enforcement orders  
egen unique_order1=tag(PCAOB_ReleaseNo) if Individual==1
```

count if unique_order1 == 1 //208 enforcement orders name one or more individuals
sum Individual //248 unique number of individuals
sum Individual if US==0 // 58 foreign individuals

***table 1 Panel B**

import excel "C:\Users\mengl\OneDrive - Temple University\Desktop\Temple\Audit Partner
AAER\JAR\Data\KLMP.xlsx", sheet("Final") firstrow clear //import the dataset with manually
collected data on individual culpable auditor's penalties, labor market, and personal outcomes

tab year SEC, column //Panel B

***table 2**

total Unethicalorimproperprofession Periodicquarterlyandannual Accountantsreports
Antifraudprovisions Auditrequirements Fraudulentinterstatetransactio Recordkeepingprovisions
Internalcontrolprovisions Other_sec //Panel A

total Evidence Rules_standards Cooperation Quality_controls Rules_filings Integrity Manipulation
Reckless Independence other_pcaob //Panel B

***table 3**

tab Big4 SEC, column //Panel A
tab partner SEC if Big4==1 , column //Panel B
tab partner SEC if Big4==0 , column //Panel B

tab Male SEC if Big4==1 , column //Panel C
tab Male SEC if Big4==0 , column //Panel C

gen group=1 if SEC==1&Big4==1
replace group=2 if SEC==0&Big4==1
replace group=3 if SEC==1&Big4==0
replace group=4 if SEC==0&Big4==0

sort group
by group: sum age //Panel D

sort Big4
by Big4: tab license_no_dummy SEC, column // Panel E 1- not licensed 2-licensed in 1 state 3-licensed
in 2 states 4-licensed in more than 2 states

***table 4**

sort group
by group: tab Suspension_dummy //Panel A 1-no suspension 2-up to 2 years 3-3-7 years 4-permanent
by group: summarize CPI_fines if fines_dummy ==1, detail //Panel B&C
by group: summarize CPI_disgorge if disgorge_dummy ==1, detail //Panel B

***table 5**

sort Big4
by Big4: tab state_board_outcome SEC if CPA_Dummy==1, column //Panel A 1-no action 2-
probation 3-restriction 4-suspension 5-revocation 6-no record
by Big4: tab state_board_fine_dummy SEC if CPA_Dummy==1&state_board_outcome> 1, column
//Panel B

sort group
by group: sum CPI_Fine_state //Panel B

***table 6**

replace left_firm=2 if missing(left_firm) //1-leave firm 0 - do not leave firm 2- unknowns
tab left_firm SEC if Big4==1, column //Panel A
tab left_firm SEC if Big4==0&SinglePartnerFirm==1, column //Panel B
tab left_firm SEC if Big4==0&SinglePartnerFirm==0, column //Panel B

egen group1=concat(left_firm SEC)
sort group1
by group1: sum age if Big4==1 //Panel A
by group1: sum age if Big4==0&SinglePartnerFirm==1 //Panel B
by group1: sum age if Big4==0&SinglePartnerFirm==0 //Panel B

sort group
replace new_job_dummy=2 if age>54 &new_job_dummy==3 &left_firm==1 //any individual left
firm but subsequent employment unknown and 55 or above are considered retired

tab new_job_dummy SEC if left_firm==1&Big4==1, column //Panel A 1- jail/left us 2-retired 3-
unknown 4-continue working
tab new_job_dummy SEC if left_firm==1&Big4==0, column //Panel B

egen group2=concat(new_job_dummy SEC)
sort group2
by group2: sum age if left_firm==1&Big4==1 //Panel A
by group2: sum age if left_firm==1&Big4==0 //Panel B

***table 7**

sort group
tab new_partner_dummy SEC if
Big4==1&NewEmployerSinglePartnerFirm==1&new_job_dummy==4&left_firm==1 //Panel A
tab new_partner_dummy SEC if
Big4==1&NewEmployerSinglePartnerFirm==0&new_job_dummy==4&left_firm==1 //Panel A
tab new_position_co SEC if Big4==1&new_job_dummy==4&left_firm==1, column // Panel A 1-
board member, 2 CEO 3 CFO 4 COO 5 other c-suite 6 other

tab new_partner_dummy SEC if
Big4==0&NewEmployerSinglePartnerFirm==1&new_job_dummy==4&left_firm==1 //Panel C
tab new_partner_dummy SEC if
Big4==0&NewEmployerSinglePartnerFirm==0&new_job_dummy==4&left_firm==1 //Panel C
tab new_position_co SEC if Big4==0&new_job_dummy==4&left_firm==1, column //Panel C

by group:sum market_cap if publiclylisted==1 //Panel A&C

by group: tab relocation if new_job_dummy==4&left_firm==1 //Panel B&D
total relocation_local relocation_50_miles if group==1&new_job_dummy==4&left_firm==1 //Panel
B
total relocation_local relocation_50_miles if group==2&new_job_dummy==4&left_firm==1 //Panel
B
total relocation_local relocation_50_miles if group==3&new_job_dummy==4&left_firm==1 //Panel
D
total relocation_local relocation_50_miles if group==4&new_job_dummy==4&left_firm==1 //Panel
D

***table 8**

tab group property_sale_dummy //Panel A &B 0- no sale, 1 primary only, 2 secondary only 3 both
primary and secondary
tab group replacement_dummy, row //Panel A &B
tab group negative_cash_flow if property_sale_dummy>0&propertypurchase>0, row //Panel A and
B big4_SEC has one unknown for cash flow calculation
by group: sum Buyselldifference if property_sale_dummy>0&propertypurchase>0, detail //Panel A
and B
tab group default_dummy, row //Panel A and B

***online appendix 1 tables 1, 2, and 3**

*These tables repeat the tabulated descriptive statistics from Tables 6, 7, and 8 for culpable auditors. It also presents similar descriptive information for the comparison auditor sample. Please refer to Part II of this file below.

***online appendix 2 table1**

***This table presents variable definitions**

***online appendix 2 table2**

gen depart=1 if left_firm==1
replace depart=0 if left_firm==0

gen indiv_suspen=log(1+Suspension_bar_period)
gen indiv_penalty=log(1+CPI_fines)
gen indiv_disgorge=log(1+CPI_disgorge)
gen firm_suspen=log(1+Firm_Suspension)
gen firm_penalty=log(1+CPI_firm_penalty)
gen firm_disgorge=log(1+CPI_firm_disgorge)
gen stay_pa=public_accounting

est clear

eststo:logit depart Big4 partner violation SEC indiv_suspen indiv_penalty indiv_disgorge age
firm_suspen firm_penalty firm_disgorge i.year, robust //note: 16 obs dropped were added to number
of observations

eststo:logit stay_pa Big4 partner violation SEC indiv_suspen indiv_penalty indiv_disgorge age
firm_suspen firm_penalty firm_disgorge i.year if depart==1, robust

esttab using klmplOA3.csv, replace b(3) starlevels(* 0.10 ** 0.05 *** 0.01) nogap pr2

***online appendix 3**

sort group
egen groupoa3=concat(group Suspension_dummy) //1-no suspension 2-up to 2 years 3-3-7 years 4-
permanent

sort groupoa3

by groupoa3: sum age CPI_fines

sort group

by group: tab Suspension_dummy state_board_outcome if CPA_Dummy==1, row //1-no action 2-
probation 3-restriction 4-suspension 5-revocation 6-no record

by group: tab Suspension_dummy CPA_Dummy //non_CPAs are included in the other/missing data
category

by group: tab Suspension_dummy left_firm, row //1-leave firm 0 - do not leave firm 2- unknowns, do
not leave firm and unknowns are grouped together for table reporting

by group: tab Suspension_dummy new_job_dummy if left_firm==1, row // 1- jail/left us 2-retired 3-unknown 4-continue working, first three categories are grouped together for table reporting
 by group: tab Suspension_dummy public_accounting if left_firm==1&new_job_dummy==4
 by group: tab Suspension_dummy property_sale_dummy //0- no sale, 1 primary only, 2 secondary only 3 both primary and secondary, last three categories are grouped together for table reporting
 by group: tab Suspension_dummy LNnotavailable //reported as missing data in the last row of the table
 by group: tab Suspension_dummy replacement_dummy
 by group: tab Suspension_dummy default_dummy, row

***online appendix 4**

```
egen groupoa4=concat(group FirmNamed)
sort groupoa4
by groupoa4: sum age CPI_fines
by groupoa4: tab Suspension_dummy //1-no suspension 2-up to 2 years 3-3-7 years 4-permanent
by groupoa4: sum Suspension_bar_period if Suspension_dummy<4 //average suspension in years for those not permanently suspended
sort group
by group: tab FirmNamed left_firm, row
by group: tab FirmNamed new_job_dummy if left_firm==1, row // 1- jail/left us 2-retired 3-unknown 4-continue working, first three categories are grouped together for table reporting
by group: tab FirmNamed public_accounting if new_job_dummy==4&left_firm==1
by group: tab FirmNamed property_sale_dummy //0- no sale, 1 primary only, 2 secondary only 3 both primary and secondary, last three categories are grouped together for table reporting
by group: tab FirmNamed LNnotavailable //reported as missing data in the last row of the table
by group: tab FirmNamed replacement_dummy
by group: tab FirmNamed default_dummy, row
```

***online appendix5**

```
replace Litigation=0 if missing(Litigation) //no litigation
egen groupoa5=concat(group Litigation)
sort groupoa5
by groupoa5: sum age CPI_fines
by groupoa5: tab Suspension_dummy //1-no suspension 2-up to 2 years 3-3-7 years 4-permanent
by groupoa5: sum Suspension_bar_period if Suspension_dummy<4 //average suspension in years for those not permanently suspended
sort group
by group: tab Litigation left_firm, row
by group: tab Litigation new_job_dummy if left_firm==1, row // 1- jail/left us 2-retired 3-unknown 4-continue working, first three categories are grouped together for table reporting
by group: tab Litigation public_accounting if new_job_dummy==4&left_firm==1
by group: tab Litigation property_sale_dummy //0- no sale, 1 primary only, 2 secondary only 3 both primary and secondary, last three categories are grouped together for table reporting
by group: tab Litigation LNnotavailable //reported as missing data in the last row of the table
by group: tab Litigation replacement_dummy
```

by group: tab Litigation default_dummy, row

PART II

Data on Comparison Auditors

We use Stata to collate Revelio files and clean the dataset. We then use SAS to further modify the dataset and tabulate descriptive data.

1. Stata Code

```
/******  
  
*****  
***** This code is used to clean the raw data imported from Revelio. *****  
*****  
*/  
/******  
**  
1. Transferring position dataset to stata files. We restrict the observations  
from United States (Log Skipped)  
*****  
*/  
local files : dir "`filepath'" files "*.csv"  
di ``files''  
  
clear  
local filenames: dir "." files "academic_individual_position_csv*.csv"  
  
foreach f of local filenames {  
    import delimited using "`f'", varnames(1) clear  
        keep if country == "United States"  
        keep user_id company_name state startdate enddate title_raw  
  
        destring user_id, generate(user_id_new)  
        format user_id_new %12.0g  
save "m_`f'.dta", replace  
}  
  
/******  
**  
2. From the above Stata files, we only keep observations that ever work  
at top 100 accounting firms. We fuzzy match based on firm names (Log Skipped)  
*****  
*/  
  
clear  
local filenames: dir "." files "m_academic_individual_position_csv*.csv.dta"
```

```

foreach f of local filenames {
use "`f'"

gen company_name1 = subinstr(company_name, ".", "", .)
replace company_name1 = subinstr(company_name1, ",", "", .)
replace company_name1 = subinstr(company_name1, "&", "", .)
replace company_name1 = subinstr(company_name1, "+", "", .)
replace company_name1 = subinstr(company_name1, "(", "", .)
replace company_name1 = subinstr(company_name1, ")", "", .)
replace company_name1 = subinstr(company_name1, "/", "", .)
replace company_name1 = subinstr(company_name1, "|", "", .)
replace company_name1 = subinstr(company_name1, "-", "", .)
replace company_name1=lower(company_name1)

save "`f'", replace

matchit user_id_new company_name1 using ipa_070424.dta , idusing(No) txtusing(firm)
sim(token) gen(score)

bysort user_id_new (score): keep if score==score[_N] & score > 0.8

sort user_id_new company_name1
quietly by user_id_new company_name1: gen dup = cond(_N==1,0,_n)
drop if dup>1
merge 1:m user_id_new company_name1 using "`f'"
keep if _merge ==3
save "i_`f'.dta", replace

}

/*****
**
3. Merge all the files from the above step (Log Skipped)
*****/
*/

local filenames : dir "." files "i_m_academic_individual_position_csv*.dta"

tempfile building
save `building', emptyok

foreach f of local filenames {
use "`f'", clear
append using `building'

```

```

save `building', replace
}
/*****
**
4. Create two files from the above step because one big file won't get
imported into SAS
*****/
*/
log using JAR_KLMP

gen obs_num = _n

* Create the first subset (first half)
keep if obs_num <= 1185359
save part_a_1.dta, replace

* Reload the original dataset
use "C:\Users\micalle0425\Desktop\part_a.dta"

* Create the second subset (second half)
gen obs_num = _n

keep if obs_num > 1185359
save part_a_2.dta, replace

/*****
**
SAS LOG FILE STARTS HERE 1

*****/
*/
/*****
**
5. Put all different job titles into Partner, Manager, and Staff
*****/
*/
* Load title.dta file exported from SAS

use "C:\Users\micalle0425\Desktop\title.dta"

replace title_raw = strlower(title_raw)
gen title_raw1 = substr(title_raw, " ", "", .)

```

```

keep if strpos(title_raw1, "assurance") | strpos(title_raw1, "audit") | strpos(title_raw1,
"accountant")

gen flag=0
replace flag=1 if strpos(title_raw1, "accountant") | strpos(title_raw1, "auditor") |
strpos(title_raw1, "auditstaff") | strpos(title_raw1, "senioraccountant") | strpos(title_raw1,
"seniorassociate") | strpos(title_raw1, "staff") | strpos(title_raw1, "associate") | strpos(title_raw1,
"senior")

replace flag=2 if strpos(title_raw1, "auditmanager") | strpos(title_raw1, "manager") |
strpos(title_raw1, "seniormanager")

replace flag=3 if strpos(title_raw1, "owner") | strpos(title_raw1, "partner") | strpos(title_raw1,
"principal") | strpos(title_raw1, "shareholder")

save title_filtered.dta
/*****
**
SAS LOG FILE STARTS HERE 2

*****/

/*****
**
6. To Obtain Departing Comparison Individuals' Full Employment Information
*****/

clear
local filenames: dir "." files "m_academic_individual_position_csv*.csv.dta"

foreach f of local filenames {
use "`f'"

keep user_id_new state startdate enddate title_raw company_name

merge m:1 user_id_new using unique_userid_turnover.dta
keep if _merge ==3
save "b_`f'.dta", replace
}

```

```

/*Merge all the files*/
local filenames : dir "." files "b_m_academic_individual_position_csv*.dta"

tempfile building
save `building', emptyok

foreach f of local filenames {
  use "`f'", clear
  append using `building'

save `building', replace
}

save partb.dta

/*****
**
SAS LOG FILE STARTS HERE 3
*****/
*/
/*****
**
7. Make the company name matchable to 100 accounting firms
*****/
*/
use "C:\Users\micalle0425\Desktop\partb4.dta"

gen company_name1 = substr(company_name, ".", "",.)
replace company_name1 = substr(company_name1, ",", "",.)
replace company_name1 = substr(company_name1, "&", "",.)
replace company_name1 = substr(company_name1, "+", "",.)
replace company_name1 = substr(company_name1, "(", "",.)
replace company_name1 = substr(company_name1, ")", "",.)
replace company_name1 = substr(company_name1, "/", "",.)
replace company_name1 = substr(company_name1, "|", "",.)
replace company_name1 = substr(company_name1, "-", "",.)
replace company_name1=lower(company_name1)

save partb_v2.dta

/*****
**
SAS LOG FILE STARTS HERE 4

```

```

*****
*/
/*****
**
8. Fuzzy match company names against company names in Compustat
*****
*/
use "C:\Users\micalle0425\Desktop\no_match.dta"

matchit row_num company_name1 using funda_name.dta , idusing(gvkeyn)
txtusing(new_company_name) sim(token) gen(score)
bysort row_num (score): keep if score==score[_N] & score > 0.8

save funda_matched.dta

/*****
**
SAS LOG FILE STARTS HERE 5
*****
*/
/*****
**
9. For the real estate table,
1) grab data from Revelio User files and merge it to our data used for Step 3 above

get all control individulas from the data used
for step 3 above. Then we expand the search by not limiting to
'audit', 'assurance', or 'accountant'.

*****
*/
local files : dir "`filepath'" files "*.csv"
di ``files'''

clear
local filenames: dir "." files "academic_individual_user_csv*.csv"

foreach f of local filenames {
import delimited using "`f'", varnames(1) clear
keep if user_country == "United States"
keep user_id firstname lastname fullname sex_predicted profile_linkedin_url
user_location
destring user_id, generate(user_id_new)
format user_id_new %12.0g

```

```

merge 1:m user_id_new using part_a.dta
keep if _merge ==3

gen location=substr(user_location, strpos(user_location, ",") + 1, length(user_location))
gen CPASStateCountry=substr(location, 1, strpos(location, ",") - 1)
drop if missing(CPASStateCountry)
replace title_raw = subinstr(title_raw, " ", "", .)
replace title_raw = strlower(title_raw)
replace CPASStateCountry= subinstr(CPASStateCountry, " ", "", .)
replace CPASStateCountry = strlower(CPASStateCountry)

gen wanted=1 if inlist(title_raw, "partner", "principal", "owner", "seniormanager", "manager",
"director", "staffaccountant")
replace wanted=1 if inlist(title_raw, "auditstaff", "senioraccountant", "consultant", "shareholder",
"accountant", "auditor", "seniorassociate", "senior", "taxmanager")
keep if wanted==1
gen title_raw1=title_raw

save "m_`f'.dta", replace
}

local filenames : dir "." files "m_academic_individual_user_csv*.dta"

tempfile building
save `building', emptyok

foreach f of local filenames {
    use "`f'", clear
    append using `building'

save `building', replace
}
/*****
**
SAS LOG FILE STARTS HERE 6
*****/

/*****
**
10. Get bachelor year data (Log Skipped)
*****/
local files : dir "`filepath'" files "*.csv"

```

```
di ``files``
```

```
clear
```

```
local filenames: dir "." files "academic_individual_user_education_csv*.csv"
```

```
foreach f of local filenames {  
    import delimited using "`f'", varnames(1) clear  
    destring user_id, generate(user_id_new)  
    format user_id_new %12.0g
```

```
merge 1:m user_id_new using part_c.dta  
keep if _merge ==3  
save "m_`f'.dta", replace  
}
```

```
foreach f of local filenames {  
    use "`f'", clear  
    append using `building'
```

```
save `building', replace  
}
```

2. SAS Code

```
libname a "C:\Users\micalle0425\Desktop\conditional_klmp";
libname b "C:\Users\micalle0425\Desktop\JAR_KLMP\JAR";

/*****
***

SAS LOG FILE STARTS HERE 1: To Calculate Turnover Rate for the Comparison
Auditors

*****/
PROC import datafile="C:\Users\micalle0425\Desktop\part_a_1.dta"
out=a.part_a_1
dbms=STATA
replace;
run;
PROC import datafile="C:\Users\micalle0425\Desktop\part_a_2.dta"
out=a.part_a_2
dbms=STATA
replace;
run;

data a.part_a;
set a.part_a_1 a.part_a_2;
run;

/*****
***

STATA LOG FILE STARTS HERE 2: To Put all Different Job Titles into Partner,
Manager, and Staff
Do this on Stata: Step 5

*****/
proc sort data=a.part_a nodupkey out=a.title;
by title_raw;
run;

PROC export DATA= a.title
OUTFILE= "C:\Users\micalle0425\Desktop\title.dta"
          DBMS=STATA REPLACE;
RUN;

/*****
***

SAS LOG FILE STARTS HERE 2

*****/
```

```

PROC import datafile="C:\Users\micalle0425\Desktop\title_filtered.dta"
out=a.title_filtered
dbms=STATA
replace;
run;

proc sort data=a.title_filtered;
by title_raw1 descending flag;
run;

proc sort data=a.title_filtered nodupkey;
by title_raw1;
run;

data a.part_a;
set a.part_a;
title_raw=lowercase(title_raw);
title_raw1=compress(title_raw, '');
run;

proc sql;
create table a.part_a1 as
select a.*, b.flag
from a.part_a a left join a.title_filtered b
on a.title_raw1=b.title_raw1;
quit;

proc sort data=a.part_a1;
by flag;
run;

/*Make a dummy varaible for those observations working for Big 4*/
data a.part_a2;
set a.part_a1;
if flag ne '';
if flag ne 0;
if company_name1 = "deloitte and touche" or company_name1 = "deloitte llp new
york" or company_name1="ernst young global ltd" or
company_name1="ernst and young" or company_name1="ernst young" or
company_name1 = "kpmg llp" or company_name1="kpmg llp canada" or
company_name1="kpmg huazhen llp"
or company_name1 ="pricewaterhousecoopers ireland llp" or
company_name1="pricewaterhousecoopers llp" or company_name1=
"pricewaterhousecoopers llp singapore"
or company_name1= "pwc" or company_name1="pwc law llp" then
big4_comparison=1; else big4_comparison=0;
state=lowercase(state);
state=compress(state, '');
run;

/*This dataset is the master file that has all of our sampled culpable
auditors*/
PROC import datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\match partner
list.dta"
out=a.partner_list

```

```

dbms=STATA
replace;
run;

/*This masterfile has typos for culpable auditor no=28, 29, 30, 31, 32,
424,103, and 278*/
/*Example, for No 28 partner, we mistakenly had "parner" when we created the
data manually*/
data a.partner_list;
set a.partner_list;
if no=28 then title_raw1="partner";
if no=29 then title_raw1="partner";
if no=30 then title_raw1="partner";
if no=31 then title_raw1="partner";
if no=32 then title_raw1="partner";
if no=424 then title_raw1="staffaccountant";

if no=103 then title_raw1="partner";
if no=278 then title_raw1="manager";

if title_raw1="owner" then title_raw1="partner";
run;

proc sort data=a.partner_list nodupkey out=a.test;
by title_raw1;
run;
/*Put all different job titles into Partner, Manager, and Staff*/
data a.partner_list;
set a.partner_list;
if title_raw1="accountant" or title_raw1="auditor" or title_raw1="auditstaff"
or title_raw1="consultant" or
title_raw1="senior" or title_raw1="senioraccountant" or
title_raw1="seniorassociate" or title_raw1="staff" or
title_raw1="staffaccountant" then title_flag=1;
else if title_raw1="auditmanager" or title_raw1="manager" or
title_raw1="seniormanager" or title_raw1="taxmanager" or
title_raw1="director" then title_flag=2;
else if title_raw1="owner" or title_raw1="partner" or title_raw1="principal"
or title_raw1="shareholder" then title_flag=3;
run;

/*Merge all the control individuals that ever work at top 100 accounting
firms based on 'state' and job title*/
proc sql;
create table a.partner_list1 as
select a.*, b.user_id new, b. company_name1, b. state, b. startdate, b.
enddate, b. title_raw1 as title_raw1_control, b. big4_comparison
from a.partner_list a left join a.part_a2 b
on a.CPASTateCountry=b.state
and a.title_flag = b. flag ;
quit;

/*Match if the control individuals have the same Big 4 membership as the
matched culpable auditors*/
/*obs: 386,683*/
data a.partner_list2;
set a.partner_list1;

```

```

if startdate ne '';
end_year = substr(enddate, 1, 4);
end_yearn=end_year*1;
if end_yearn='' then end_yearn=2024;
start_year=substr(startdate, 1, 4);
start_yearn=start_year*1;

if big4=big4_comparison;
run;

proc sort data=a.partner_list2;
by no user_id_new;
run;

/*Removing the (weird) duplicate control individuals having the same title
working at the same firm with the same start and end year*/
/*obs: 385,075*/
proc sort data=a.partner_list2 nodupkey out=a.partner_list3;
by no user_id_new start_yearn end_yearn company_name1 title_raw1;
run;

proc sort data=a.partner_list3;
by no user_id_new startdate;
run;

/*Create turnover dummy*/
/*Need to make sure control individuals work at the firm in t-2 where t is
the enforcement year*/
data a.partner_list4;
set a.partner_list3;
if start_yearn<=enforcement_year-2<end_yearn;
if (start_yearn<=enforcement_year) and (enforcement_year-
2<=end_yearn<=enforcement_year+1) then turnover=1; else turnover=0;
run;
/*only keeping the next subsequent job information*/
proc sort data=a.partner_list4;
by no user_id_new company_name1 descending enddate;
run;
proc sort data=a.partner_list4 nodupkey out=a.partner_list4;
by no user_id_new;
run;

/*We need to make sure they depart to some other firms rather than staying at
the same firm to calculate departure rate*/
/*It is possible that one individual can have two different rows while the
employer is the same. This can happen for example because their title
changes*/
/*For turnover rate, we cannot count as turnover if they change their title
at the same firm*/
data a.rubbish (keep= no user_id_new company_name1 startdate enddate) ;
set a.partner_list4;
if turnover=1;
run;

proc sql;
create table a.rubbish1 as

```

```

select a.*, b. company_name1 as rubbish_company_name1, b. startdate as
rubbish_startdate, b. enddate as rubbish_enddate
from a.rubbish a left join a.partner_list3 b
on a.no=b.no
and a.user_id_new = b. user_id_new;
quit;

data a.rubbish2;
set a.rubbish1;
if rubbish_enddate = '' then rubbish_enddate="2024-12-31";

    enddate_ch = input(enddate, yymmdd10.);
    format enddate_ch date9.;

    rubbish_startdate_ch = input(rubbish_startdate, yymmdd10.);
    format rubbish_startdate_ch date9.;
run;

data a.rubbish3;
set a.rubbish2;
if enddate<=rubbish_enddate;
if startdate=rubbish_startdate and enddate=rubbish_enddate then delete;
run;

proc sort data=a.rubbish3 nodupkey;
by no user_id_new;
run;

data a.rubbish3;
set a.rubbish3;
if company_name1=rubbish_company_name1 and
enddate_ch<=rubbish_startdate_ch<=enddate_ch+180 then no_turnover=1; else
no_turnover=0;
run;

proc means data=a.rubbish3;
var no_turnover;
run;

proc sql;
create table a.partner_list5 as
select a.*, b. no_turnover
from a.partner_list4 a left join a.rubbish3 b
on a.no=b.no
and a.user_id_new = b. user_id_new
and a. startdate = b.startdate
and a. enddate = b.enddate;
quit;

data a.partner_list6;
set a.partner_list5;
if no_turnover='' then no_turnover=0;
if turnover=1 and no_turnover=1 then turnover=0;
run;

proc freq data=a.partner_list6; tables no / noprint out=a.checkdup; run;

```

```

proc means data=a.partner_list6;
var turnover;
run;

proc sort data=a.partner_list6;
by no user_id_new descending turnover;
run;

proc sort data=a.partner_list6 nodupkey out=a.partner_list8;
by no user_id_new;
run;

/*****
***

We use a.partner_list8 for Calculating Turnover Rates for Comparison
Individuals
Online Appendix 1 Table 1 Panel A

*****/
/*Calculate control individual turnover that is matched to Big 4 SEC
enforcement culpable auditors*/
data big4_aaer;
set a.partner_list8;
if big4=1;
if aaer=1;
run;

proc sort data=big4_aaer;
by no;
run;

proc means data=big4_aaer noprint;
by no;
var turnover;
output out=big4_mean n=frequency mean=mean_turnover;
run;

proc means n mean data=big4_mean;
var mean_turnover;
run;

/*Calculate control individual turnover that is matched to Big 4 PCAOB
enforcement culpable auditors*/
data big4_pcaob;
set a.partner_list8;
if big4=1;
if aaer=0;
run;

proc sort data=big4_pcaob;
by no;
run;

proc means data=big4_pcaob noprint;
by no;

```

```

var turnover;
output out=big4_mean n=frequency mean=mean_turnover;
run;

proc means n mean data=big4_mean;
var mean_turnover;
run;

/*****
***

We use a.partner_list8 for Calculating Turnover Rates for Comparison
Individuals
Online Appendix 1 Table 1 Panel B

*****/
**/
/*Calculate control individual turnover that is matched to Non-Big 4 SEC
enforcement culpable auditors*/
data n_big4_sec;
set a.partner_list8;
if big4=0;
if aaer=1;
run;

proc sort data=n_big4_sec;
by no;
run;

proc means data=n_big4_sec noprint;
by no;
var turnover;
output out=n_big4_mean n=frequency mean=mean_turnover;
run;

proc means n mean data=n_big4_mean;
var mean_turnover;
run;

/*Calculate control individual turnover that is matched to Non-Big 4 PCAOB
enforcement culpable auditors*/
data n_big4_pcaob;
set a.partner_list8;
if big4=0;
if aaer=0;
run;

proc sort data=n_big4_pcaob;
by no;
run;

proc means data=n_big4_pcaob noprint;
by no;
var turnover;
output out=n_big4_mean n=frequency mean=mean_turnover;
run;

```

```

proc means n mean data=n_big4_mean;
var mean_turnover;
run;

/*****
***

Below Code is Used for Exploring the Labor Market Outcomes for Comparison
Individuals

*****/
**/
/*Start from those that we identify as departing the firm*/
data a.partb;
set a.partner_list8;
if turnover=1;
run;

/*To get those departing individuals' full employment information, we export
unique user_id and merge full job history*/
proc sort data=a.partb nodupkey out=a.unique_userid_turnover;
by user_id_new;
run;

data a.unique_userid_turnover (keep=user_id_new);
set a.unique_userid_turnover;
run;

PROC export DATA= a.unique_userid_turnover
OUTFILE= "C:\Users\micalle0425\Desktop\unique_userid_turnover.dta"
        DBMS=STATA REPLACE;
RUN;

/*****
***

STATA LOG FILE STARTS HERE 3: To Obtain Departing Comparison Individuals'
Full Employment Information
Do this on Stata: Step 6

*****/
**/

/*****
***

Start Again Here After Completing Step 6 from Stata
/*Partb.dta has all the information about jobs for comparison individuals
departing the firm*

*****/
**/

PROC import datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\Position\USA
position only\partb.dta"

```

```

out=a.partb_job
dbms=STATA
replace;
run;

data a.partb_job (drop=_merge);
set a.partb_job;
run;

data a.partb (keep=no aaer enforcement_year user_id_new turn_over_end_date
big4) ;
set a.partb;
turn_over_end_date=enddate;
run;

proc sql;
create table a.partb1 as
select a.*, b. *
from a.partb a left join a.partb_job b
on a.user_id_new=b.user_id_new;
quit;

proc sort data=a.partb1;
by no user_id_new startdate;
run;
/*We only keep the subsequent job information which is after the job we use
to identify turnover*/
/*If we can identify any job starting after the end date for the job we use
to identify turnover, this indicates that they continue to work*/
data a.partb2;
set a.partb1;
if turn_over_end_date<=startdate;
continue_working=1;
run;

/*We only grab the next subsequent job for our table*/
/*Partb3 dataset has the next subsequent job that control people take
conditional on working after departure*/
proc sort data=a.partb2 nodupkey out=a.partb3;
by no user_id_new;
run;

/*****
***

Start Again Here After Completing Step 6 from Stata
Data a.Table6 is used for documenting whether departing control individuals
continue working or retire.
That is, Online Appendix 1 Table 1 Panels A and B

*****/
proc sql;
create table a.Table6 as
select a.*, b. continue_working
from a.partb a left join a.partb3 b
on a.no=b.no

```

```

and a. user_id_new = b. user_id_new;
quit;

data a.table6;
set a.table6;
if continue_working='' then continue_working=0;
run;

data OA_Table1;
set a.table6;
if big4=0;
if aaer=0;
if continue_working=0;
run;

/*Now, we need to catagorize all different comapany names*/
data a.partb4;
set a.partb3;
if company_name ='' then company_name="other";
run;

PROC export DATA= a.partb4
OUTFILE= "C:\Users\micalle0425\Desktop\partb4.dta"
        DBMS=STATA REPLACE;
RUN;

/*****
***

STATA LOG FILE STARTS HERE 4: Make the Company Name Matchable to 100
Accounting Firms
Do this on Stata: Step 7

*****/
/*****
***

SAS LOG FILE STARTS HERE 4

*****/

PROC import datafile="C:\Users\micalle0425\Desktop\partb_v2.dta"
out=a.partb5
dbms=STATA
replace;
run;

/*First, identify if new employer is one of Big 4 audit firms*/
data a.partb5;
set a.partb5;
if company_name1 = "deloitte accounting" or company_name1 = "deloitte
corporate finance 1388185" or company_name1="deloitte development llc" or
company_name1="deloitte llp new york" or company_name1="deloitte tax" or
company_name1 = "ernst young global ltd" or company_name1="kpmg audit sarl"
or company_name1="kpmg australia pty ltd"

```

```

or company_name1 ="kpmg ireland" or company_name1="kpmg llp" or
company_name1= "kpmg llp advisory services valuations"
or company_name1= "kpmg llp canada" or company_name1= "kpmg new zealand" or
company_name1= "kpmg sa"
or company_name1= "pricewaterhousecoopers ireland llp" or company_name1=
"pricewaterhousecoopers llp" or company_name1= "pricewaterhousecoopers llp uk
broker"
or company_name1= "pricewaterhousecoopers sia" or company_name1=
"pricewaterhousecoopers spain" or company_name1= "pwc" then
new_employer_big4=1; else new_employer_big4=0;
run;

/*Second, I match based on top 100 accounting firm names*/
PROC import datafile="C:\Users\micalle0425\Desktop\ipa_070424.dta"
out=a.top100
dbms=STATA
replace;
run;

data a.top100;
set a.top100;
top100=1;
run;

proc sql;
create table a.partb6 as
select a.*, b. top100
from a.partb5 a left join a.top100 b
on a.company_name1=b.firm_1;
quit;

proc sort data=a.partb6;
by no user_id_new descending top100;
run;

proc sort data=a.partb6 nodupkey;
by no user_id_new;
run;

data a.partb6;
set a.partb6;
if top100 = '' then top100=0;
run;

data a.test;
set a.partb6;
if top100=0;
run;

proc sort data=a.test nodupkey;
by company_name1;
run;
/*Flag if the company name has llp or cpa*/
data a.test;
set a.test;
if find(company_name1, 'llp') > 0 then flag_llp = 1;
else flag_llp = 0;

```

```

        if find(company_name1, 'cpa') > 0 then flag_cpa = 1;
    else flag_cpa = 0;
run;

proc sql;
create table a.partb7 as
select a.*, b. flag_llp, b. flag_cpa
from a.partb6 a left join a.test b
on a.company_name1=b.company_name1;
quit;

data a.partb7;
set a.partb7;
if flag_llp = '' then flag_llp=0;
if flag_cpa = '' then flag_cpa=0;
if company_name1="other" then unknown=1; else unknown=0;
run;

/*Match using company names that show up in Compustat*/
data a.test;
set a.partb7;
if new_employer_big4=0 and top100=0 and flag_llp=0 and flag_cpa=0 and
unknown=0;
run;

proc sort data=a.test nodupkey;
by company_name1;
run;

data a.test (keep=company_name1 row_num enforcement_year);
set a.test;
    row_num = _N_;
run;
/*Export this dataset to match with Compustat firm names*/
PROC export DATA= a.test
OUTFILE= "C:\Users\micalle0425\Desktop\no_match.dta"
        DBMS=STATA REPLACE;
RUN;

/*This dataset has all company names and years
data b.funda_name (drop=conm);
set b.funda_name;
conm_low=lowcase(conm);
gvkeyn=gvkey*1;
run;

data b.funda_name;
set b.funda_name;
new_company_name=compress(conm_low, '-!@#$$%^&*()');
run;

data b.funda_name1 (drop=conm_low);
set b.funda_name;
run;

PROC export DATA= a.funda_name1

```

```

OUTFILE= "C:\Users\micalle0425\Desktop\matching company name\funda_name.dta"
      DBMS=STATA REPLACE;
RUN;*/

/*****
***

STATA LOG FILE STARTS HERE 5: Fuzzy match company names against company names
in Compustat
Do this on Stata: Step 8

*****/
**/
/*****
***

SAS LOG FILE STARTS HERE 5

*****/
**/

PROC import datafile="C:\Users\micalle0425\Desktop\funda_matched.dta"
out=a.funda_matched
dbms=STATA
replace;
run;

data a.funda_matched;
set a.funda_matched;
funda=1;
run;

proc sort data=a.funda_matched;
by new_company_name descending score;
run;

proc sort data=a.funda_matched nodupkey;
by new_company_name;
run;

proc sql;
create table a.partb8 as
select a.*, b. funda, b. gvkeyn
from a.partb7 a left join a.funda_matched b
on a.company_name1=b.company_name1;
quit;

/*klmp SAS data contains funda with gvkey, fyear, market_cap*/
proc sql;
create table a.partb9 as
select a.*, b. market_cap, b. tic, b. gvkeyn as year_gvkeyn
from a.partb8 a left join b.klmp b
on a.gvkeyn=b.gvkeyn
and a. Enforcement_year = b. fyear;
quit;

```

```

data a.partb10;
set a.partb9;
if tic ne '' then public_firm=1; else public_firm=0;
if gvkeyn ne '' and year_gvkeyn='' then gvkeyn='';
run;

/*With the remaining non-matched company names, we use excel to categorize
them*/
/*We flag if the company name has 'PC', 'LLC', or 'Accountant' then if the
observation is flagged, we manually check
if thses firms are public accounting firms*/
/*proc import datafile="C:\Users\micalle0425\Desktop\remaining no match MM
finished.xlsx"
out=a.remaining dbms=xlsx replace;
getnames=yes;
run; */

proc sql;
create table a.partb11 as
select a.*, b. accounting_firm
from a.partb10 a left join b.remaining b
on a.company_name1=b.company_name1;
quit;

data a.partb11;
set a.partb11;
if accounting_firm = '' then accounting_firm=0;
run;

data a.test;
set a.partb11;
if new_employer_big4=0 and top100=0 and flag_llp=0 and flag_cpa=0 and
unknown=0 and public_firm=0 and accounting_firm =0;
run;
/*We flag if the company name has 'PC', 'LLC', or 'Accountant' then if the
observation is flagged, we manually check
if thses firms are public accounting firms*/
/*We did this twice because we split the remaining firms between co-authors
and each of two did manual work separately*/
/*proc sort data=a.test nodupkey;
by company_name1;
run;
PROC export DATA= a.test
OUTFILE= "C:\Users\micalle0425\Desktop\remaining no match.xlsx"
DBMS=xlsx REPLACE;
RUN;

proc import datafile="C:\Users\micalle0425\Desktop\remaining no match.xlsx"
out=b.remaining_final dbms=xlsx replace;
getnames=yes;
run;

data b.remaining_final;
set b.remaining_final;
if pc_flag ne '' or llc_flag ne '' or accountant_flag ne '' then
accounting_firm_2=1; else accounting_firm_2=0;

```

```

run;
*/

proc sql;
create table a.partb12 as
select a.*, b. accounting_firm_2
from a.partb11 a left join b.remaining_final b
on a.company_name=b.company_name;
quit;

data a.partb12;
set a.partb12;
if accounting_firm_2 ='' then accounting_firm_2=0;
if new_employer_big4=1 or top100=1 or flag_llp=1 or flag_cpa=1 or
accounting_firm=1 or accounting_firm_2=1 then stay_public=1; else
stay_public=0;
run;

proc means data=a.partb12;
var stay_public;
run;

proc sort data=a.partb12;
by no;
run;

data a.partb12;
set a.partb12;
title_raw1=lowercase(title_raw);
run;
/*We use Excel to put titles into different categories*/
/*proc sort data=a.partb12 nodupkey out=a.test;
by title_raw1;
run;

data a.test (keep=title_raw1);
set a.test;
run;

PROC export DATA= a.test
          OUTFILE= "C:\Users\micalle0425\Desktop\title.xlsx"
          DBMS=xlsx REPLACE;

RUN;

proc import datafile="C:\Users\micalle0425\Desktop\title.xlsx"
out=b.goodluck dbms=xlsx replace;
getnames=yes;
run;

data b.goodluck;
set b.goodluck;
if Chief=1 and ceo = 0 and cfo=0 and coo=0 then other_chief=1; else
other_chief=0;
run;*/

proc sql;

```

```

create table a.partb13 as
select a.*, b. partner, b. ceo, b. cfo, b. coo, b. other_chief, b. board
from a.partb12 a left join b.goodluck b
on a.title_raw1=b.title_raw1;
quit;

proc means data=a.partb13;
var ceo cfo coo other_chief partner board;
run;

proc sort data=a.partb13;
by no ;
run;

/*Given that we have multiple control individuals matched to a given culpable
auditor, we randomly select one control individual for each culpable
auditor*/
proc surveyselect data=a.partb13 out=a.partb_final method=srs n=1 seed=56789;
strata no; run; /* Display the selected observations */

proc sort data=a.partb_final nodupkey;
by user_id_new;
run;

proc sort data=a.partb_final;
by no;
run;

/*On 2025/2/11, I am running the entire SAS code again to create the log
file*/
/*Given that the previous code requires a random selection and for the log
file creation purposes, I will use the old data I used for our paper*/
/*That is, I am using b.partb_final as opposed to a.partb_final*/
/*We use the below code by changing the code accordingly to tabulate labor
market outcome tables for our comparison group*/
/*****
***

We use b.partb_final:
Online Appendix 1 Table 2 Panels A&B

*****/
**/
/*Below code demonstrates how to construct OA 1 Table 1 Panel A & OA 2 Table
2 PAnel A*/

data aaer_big4;
set b.partb_final;
if aaer=1;
if big4=1;
if stay_public=0;
if public_firm=1;
/*if ceo= 0 and cfo= 0 and coo= 0 and other_chief= 0 and board=0*/
run;

```

```
/******  
***
```

Below code is for Real Estate Transactions
We start from Revelio file that only has individuals working at top 100
accounting firms with the same title as culpable auditors

```
*****  
**/  
/******  
***
```

STATA LOG FILE STARTS HERE 6: Grab Individuals on Revelio file that work at
top 100 accounting firms with the same title as culpable auditors
Do this on Stata: Step 9

```
*****  
**/  
/******  
***
```

SAS LOG FILE STARTS HERE 5&6

```
*****  
**/
```

```
PROC import  
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\User\user1.dta"  
out=a.partc_1  
dbms=STATA  
replace;  
run;  
PROC import  
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\User\user2.dta"  
out=a.partc_2  
dbms=STATA  
replace;  
run;  
PROC import  
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\User\user3.dta"  
out=a.partc_3  
dbms=STATA  
replace;  
run;
```

```
data a.partc_final;  
set a.partc_1 a.partc_2 a.partc_3;  
run;
```

```
proc sql;  
create table a.partc_final_1 as  
select a.*, b.*  
from a.partner_list a left join a.partc_final b  
on a.CPStateCountry=b. CPStateCountry  
and a.title_raw1 = b. title_raw1 ;  
quit;
```

```
/*Make a dummy variable for those observations working for Big 4*/
```

```

data a.partc_final_1;
set a.partc_final_1;
if company_name1 = "deloitte and touche" or company_name1 = "deloitte llp new
york" or company_name1="ernst young global ltd" or
company_name1="ernst and young" or company_name1="ernst young" or
company_name1 = "kpmg llp" or company_name1="kpmg llp canada" or
company_name1="kpmg huazhen llp"
or company_name1 ="pricewaterhousecoopers ireland llp" or
company_name1="pricewaterhousecoopers llp" or company_name1=
"pricewaterhousecoopers llp singapore"
or company_name1= "pwc" or company_name1="pwc law llp" then
big4_comparison=1; else big4_comparison=0;

if startdate ne '';
end_year = substr(enddate, 1, 4);
end_yearn=end_year*1;
if end_yearn='' then end_yearn=2024;
start_year=substr(startdate, 1, 4);
start_yearn=start_year*1;
run;

/*We want the control individuals to be working at the comparable firm,
state, with the same title in year t where t is the enforcement year*/
data a.partc_final_1;
set a.partc_final_1;
if start_yearn <=Enforcement_year =< end_yearn;
run;

/*We need to get their bachelor years to approximately calculate their age*/
proc sort data=a.partc_final_1 nodupkey out=test;
by user_id;
run;

PROC export DATA= test
OUTFILE= "C:\Users\micalle0425\Desktop\part_c.dta"
        DBMS=STATA REPLACE;
RUN;

/*Import all unique user_id with their bachelor information*/
PROC import
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\Education\education1_1_ma
tched.dta"
out=education1_1
dbms=STATA
replace;
run;
PROC import
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\Education\education1_2_ma
tched.dta"
out=education1_2
dbms=STATA
replace;
run;
PROC import
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\Education\education2_1_ma
tched.dta"
out=education2_1

```

```

dbms=STATA
replace;
run;
PROC import
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\Education\education2_2_ma
tched.dta"
out=education2_2
dbms=STATA
replace;
run;

PROC import
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\Education\education3_1_ma
tched.dta"
out=education3_1
dbms=STATA
replace;
run;

PROC import
datafile="C:\Users\micalle0425\Desktop\JAR_KLMP\JAR\Education\education3_2_ma
tched.dta"
out=education3_2
dbms=STATA
replace;
run;

data a.master_education;
set education1_1 education1_2 education2_1 education2_2 education3_1
education3_2;
run;

proc sort data=a.master_education;
by user_id bachelor_date;
run;

proc sort data=a.master_education nodupkey;
by user_id;
run;

proc sql;
create table a.partc_final_2 as
select a.*, b. bachelor_date
from a.partc_final_1 a left join a.master_education b
on a.user_id=b.user_id;
quit;

/*We drop if bachelor information is missing*/
data a.partc_final_2 ;
set a.partc_final_2 ;
if bachelor_date ne '';
bachelor_year = substr(bachelor_date, 1, 4);
bachelor_yearn=bachelor_year*1;
run;

proc sort data=a.partc_final_2 ;
by no;

```

```

run;

/*We calculate the age assuming they start their bachelor degree at the age
of 18*/
/*We match gender between the control auditors and culpable auditors*/
/*We match age between the control auditors and culpable auditors by
allowing plus and minus one year from the age of culpable auditor*/
/*We match Big 4 membership*/
data a.parc_final_3 ;
set a.parc_final_2 ;
comparison_age= enforcement_year -bachelor_yearn +18;
if sex_predicted="Female" then comparison_gender="F"; else
comparison_gender="M";
if gender=comparison_gender;
if big4_comparison = big4;
if age_aaer-1<=comparison_age=<age_aaer+1;
run;

/*We then randomly select 200 control individuals for 200 culpable auditors
for real estate transaction table*/
proc sort data=a.parc_final_3; by no;
run;

```