/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This SAS Program Generates two datasets:

1. MNC.dta: Parent-Level data used in the MNC analyses

2. Subsidiary.dta: Subsidiary-level data used in Subsidiary Analyses

The data generation process follows the following steps:

MNC dataset:

1. Obtain Compustat Data from WRDS

2. Calculate Compustat Segment variables

3. Calculate Compustat control variables

4. Import Form AP Data from Excel

5. Calculate MNC-level and country-level component auditor variables from Form AP

6. Import Audit Analytics Data

7. Calculate AA control variables

8. Merge MNC-level datasets and calculate additional controls

9. Export dataset to Stata dta file

Subsidiary dataset:

10. Import Orbis Subsidiary Data

11. Calculate Subsidiary-level control variables

12. Merge subsidiary data with Form AP component auditor data

13. Calculate component auditor variables and Jones (1991) abnormal accruals

14. Export to Stata dta file

Several final data steps are performed in Stata (see file "2. Sample Finalization.do"), including

dropping unused observations and aggregating subsidiary-level variables for use in the MNC analyses.

Datasets in SAS are stored in the "Caudit" library. All other file paths (e.g., to source data) have been replaced with "filepath"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MNC Dataset Generation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Step 1. Obtain Compustat Data from WRDS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This file collects the following datasets from WRDS Compustat Module:

1. Fundamentals Annual (funda)

2. Names (for SIC code)

3. segmerged (for Segments data)

and calculates the following variable:

1. age (number of years firm is listed on Compustat, used to calculate Firm Age)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

DM "Output; Clear; Log; Clear";

option nolabel;

**PROC** **DATASETS** LIB=work KILL;

options helpbrowser=sas;**run**;**quit**;

%let wrds=wrds.wharton.upenn.edu 4016; \* logs in to WRDS ;

options comamid=TCP remote=WRDS;

signon username=\_prompt\_ password =\_prompt\_; \* will prompt for WRDS username and password ;

rsubmit; \* rsubmit code required to que WRDS on actionable code ;

**data** A1;

set comp.funda (keep = gvkey xpr datadate fyear indfmt consol popsrc datafmt tic cusip conm fyr apdedate fdate pdate act aqc aqs at

capx ceq che csho dlc dltis dltt ebit emp ib invt lct lt mrct mrcta ni oancf oibdp pidom pifo ppent pstk re

rect sale sstk teq txp wcap xint xpr cik fic sich prcc\_c mkvalt prcc\_f au auop auopic oiadp ivao ivst epsfi

epsfx epspi epspx xidoc dp auop

pi ppegt txpd txt);\* sets Fundamentals dataset. List variables to keep ;

where datafmt = 'STD' and

indfmt='INDL' and

datafmt='STD' and

popsrc='D' and

consol='C' and

fic='USA';

format datadate mmddyy10.;

year = year(datadate);

if year < **2010** then delete;

if (at <= **0** or at = **.**) then delete; \* Delete companies with negative total assets;

if datadate = **.** then delete;

**run**;

\* Compute firm Age;

**data** age;

set comp.funda (keep = gvkey datadate);

format datadate mmddyy10.;

year = year(datadate);

**run**;

**proc** **sort** data=age nodupkey; by gvkey year; **run**;

**data** age2;

set age;

age+**1**;

by gvkey;

if first.gvkey then age = **1**;

**run**;

\* Compustat Names file for SIC code data ;

**data** A2 (keep = gvkey sic);

set comp.names;

**run**;

**proc** **sort** data=A1; by gvkey; **run**;

**proc** **sort** data=A2; by gvkey; **run**;

\* Merge fundamentals data with SIC code data;

**data** A3;

merge A1 (in=xx) A2 (in=yy);

by gvkey;

**run**;

\* Merge fundamentals data with firm age data;

**data** A3a;

merge A3 (in=xx) age2(in=yy);

by gvkey year;

if xx;

**run**;

\* Delete duplicate entries ;

**proc** **sort** data=A3a nodupkey; by gvkey year; **run**;

\* Import segments data;

**data** A4;

set comp.wrds\_segmerged;

format datadate mmddyy10.;

year = year(datadate);

if year < **2012** then delete;

if datadate = **.** then delete;

if datadate ne srcdate then delete;

**run**;

\* Download data to Caudit library;

**proc** **download** data=work.A3a out=Caudit.comp\_data1; **run**;

**proc** **download** data=work.A4 out=Caudit.comp\_segs1; **run**;

endrsubmit;

signoff; **run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 2. Calculate Compustat segments variables

Variables used include the following, by firm-year:

1. sumforsales : The sum of all non-domestic geographic segments sales

2. geosegtot : Total number of geographic segments disclosed

3. bussegtot : Total number of business segments disclosed

4. forsegtot : Total number of foreign geographic segments disclosed

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Sort the Compustat Segments Data and drop duplicates

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data=Caudit.Comp\_segs1 out=s0 nodupkey; by gvkey year snms; **run**;

\* Calculate Foreign Sales, and number of segments;

**data** S1;

SET s0;

by gvkey year;

if stype = "GEOSEG" and geotp = **3** then DO; ForSales = sales; forseg = **1**; END; else DO; ForSales = **0**; forseg=**0**; END;

if stype = "BUSSEG" then busseg = **1**; else busseg = **0**;

if stype = "GEOSEG" THEN geoseg = **1**; else geoseg = **0**;

**run**;

\*Calculate total Foreign Sales, Business Segments, and Geographic Segments;

**PROC** **MEANS** noprint sum data=S1;

by gvkey year;

var ForSales busseg geoseg forseg;

output out=S2 sum = sumforsales bussegtot geosegtot forsegtot;

**run**;

\* Merge with Compustat Fundamentals Annual Data;

**proc** **sort** data=S2 nodupkey; by gvkey year; **run**;

**proc** **sort** data=Caudit.comp\_data1 out=C0 nodupkey; by gvkey year; **run**;

**data** Caudit.Compmerged;

merge C0 (in=xx) S2(in=yy);

by gvkey year;

if xx;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 3. Calculate Compustat Control Variables

Calculate and Winsorize the following variables (see paper Appendix for variable definitions):

N Variable Name Label in Paper

1. MNC\_Size MNC Size

2. MNC\_bus\_seg MNC Business Segments

3. MNC\_geo\_seg MNC Geographic Segments

4. MNC\_foreign\_operations MNC Foreign Operations

5. MNC\_forsale MNC Foreign Sales

6. MNC\_negative\_income MNC Negative Income

7. MNC\_leverage MNC Leverage

8. MNC\_external\_financing MNC External Financing

9. MNC\_extreme\_growth MNC Extreme Growth

10. MNC\_cap\_intensity MNC Capital Intensity

11. MNC\_inv\_rec MNC Inv-Rec

12. MNC\_firm\_age MNC Firm Age

13. MNC\_lnforseg MNC For. Segments

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data=Caudit.compmerged; by gvkey year; **run**;

**data** C0;

set Caudit.compmerged;

\* Generate Control Variables;

\* Size - calculated above;

MNC\_size = log(at);

\* Number of Segments (calculated in Step 2 above, renaming here);

MNC\_bus\_seg = bussegtot;

if MNC\_bus\_seg = **.** then MNC\_bus\_seg = **0**;

MNC\_geo\_seg = geosegtot;

if MNC\_geo\_seg = **.** then MNC\_geo\_seg = **0**;

\* Foreign segments (for Subsidiary analyses);

MNC\_lnforseg = log(forsegtot + **1**);

if MNC\_lnforseg = **.** then MNC\_lnforseg = **0**;

\*Indicator for foreign operations (based on pre-tax income);

if pifo ne **.** and pifo ne **0** then MNC\_foreign\_operations = **1**; else MNC\_foreign\_operations = **0**;

\*SUM of foreign sales (per compustat segments file - caluclated in Step 2 as sumforsales);

MNC\_forsale = log(sumforsales);

if MNC\_forsale = **.** then MNC\_forsale = **0**;

\* Negative income - net income is < 0;;

if ni < **0** then MNC\_negative\_income = **1**;else MNC\_negative\_income = **0**;

if ni = **.** then MNC\_negative\_income = **.**;

\* Leverage - ratio of debt to total assets;

if dlc = **.** then dlc = **0**;

if dltt = **.** then dltt = **0**;

MNC\_leverage = (dltt + dlc) / at;

\* External financing - indicator for 10% growth in number of shares outstanding;

lagcsho = lag(csho);

if gvkey ne lag(gvkey) or lag(year) ne year-**1** then lagcsho = **.**;

share\_growth = csho / lagcsho - **1**;

if csho = **.** or lagcsho = **.** then share\_growth = **.**;

if share\_growth > **0.10** then MNC\_external\_financing = **1**; else MNC\_external\_financing = **0**;

\* Cap-intensity - ratio of net PPE to total assets;

MNC\_cap\_intensity = ppent / at;

\* INV-REC - ratio of inventory + accounts receivable to total assets;

if invt = **.** or rect = **.** then missing\_inv\_rec = **1**; else missing\_inv\_rec = **0**;

if invt = **.** then invt = **0**;

if rect = **.** then rect = **0**;

MNC\_inv\_rec = (invt + rect) / at;

\* firm-age - natural log of # of years on compustat;

MNC\_firm\_age = log(age);

\*SIC Industry Variables;

sic2 = int(sic/**100**);

sic1 = int(sic/**1000**);

\* Growth (for extreme growth calculation in next step);

growth = (sale - lag(sale)) / lag(sale);

if gvkey ne lag(gvkey) or lag(year) ne year-**1** then growth = **.**;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Calculating Extreme Growth:

Calculate avg. revenue growth by industry

Then assign extreme\_growth = 1 to top quintile

of industry-adjusted growth

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data=C0; by year sic2; **run**;

\* Calculate average sales growth by industry-year;

**proc** **means** data=C0 sum noprint;

by year sic2;

var growth;

output out=industry\_growth mean=ind\_growth;

**run**;

\* Merge industry-year growth average back in and calculate extreme growth;

**proc** **sort** data=industry\_growth; by year sic2; **run**;

**proc** **sort** data=C0; by year sic2; **run**;

**data** C1;

merge C0 (in=xx) industry\_growth (in=yy);

by year sic2;

if xx;

**run**;

**data** C2;

set C1;

growth\_indadj = growth - ind\_growth;

**run**;

\* set quintiles for industry\_adjusted growth;

**proc** **rank** data=C2

groups=**5**

out=C3;

var growth\_indadj;

ranks growth\_indadj\_quint;

**run**;

**data** C4;

set C3;

if growth\_indadj\_quint = **4** then MNC\_extreme\_growth = **1**; else MNC\_extreme\_growth = **0**;

**run**;

**proc** **sort** data=C4 nodupkey; by gvkey year; **run**;

\* Winsorize Continuous Variables;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Winsorize macro

Can winsorize or trim at specified percentiles;

dsetout = leave blank to overwrite dsetin

byvar = none for no byvar (trims/winsorizes pooled sample)

type = delete/winsor

ex: %winsor(dsetin=mydata, dsetout=mydata2, byvar=year, vars=assets earnings, pctl=0 98);

winsorizes by year at 98%, puts resulting dataset into mydata2

%winsor(dsetin=mydata, vars=assets earnings, type=delete);

trims pooled sample at 1% and 99%, puts resulting dataset back into mydata

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\* Winsorize as last step;

**%macro** winsor(dsetin=C4, dsetout = C1, byvar=none, vars=

MNC\_size MNC\_bus\_seg MNC\_geo\_seg MNC\_forsale MNC\_leverage MNC\_cap\_intensity

MNC\_inv\_rec MNC\_firm\_age MNC\_lnforseg,

type=winsor, pctl=**1** **99**);

\*nothing gets changed below\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

%if &dsetout = %then %let dsetout = &dsetin;

%let varL=;

%let varH=;

%let xn=1;

%do %until ( %scan(&vars,&xn)= );

%let token = %scan(&vars,&xn);

%let varL = &varL &token.L;

%let varH = &varH &token.H;

%let xn=%EVAL(&xn + 1);

%end;

%let xn=%eval(&xn-1);

data xtemp;

set &dsetin;

run;

%if &byvar = none %then %do;

data xtemp;

set xtemp;

xbyvar = **1**;

run;

%let byvar = xbyvar;

%end;

proc sort data = xtemp; by &byvar; run;

proc univariate data = xtemp noprint;

by &byvar;

var &vars;

output out = xtemp\_pctl PCTLPTS = &pctl PCTLPRE = &vars PCTLNAME = L H;

run;

data &dsetout;

merge xtemp xtemp\_pctl;

by &byvar;

array trimvars{&xn} &vars;

array trimvarl{&xn} &varL;

array trimvarh{&xn} &varH;

do xi = **1** to dim(trimvars);

%if &type = winsor %then %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then trimvars{xi} = trimvarl{xi};

if (trimvars{xi} > trimvarh{xi}) then trimvars{xi} = trimvarh{xi};

end;

%end;

%else %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then delete;

if (trimvars{xi} > trimvarh{xi}) then delete;

end;

%end;

end;

drop &varL &varH xbyvar xi;

run;

**%mend** winsor;

%***winsor***

**run**; \*executes the macro;

\* Save dataset;

**DATA** Caudit.C1;

SET C1;

**RUN**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 4. Import Form AP Data from Excel

File is downloaded from the PCAOB auditor search website (https://pcaobus.org/Pages/AuditorSearch.aspx)

The only edit prior to import is to break out individually identified component auditor data

into separate columns using the text-to-columns function of Excel.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**PROC** **IMPORT** OUT= Caudit.FormAP

DATAFILE= "FILEPATH\FormAP\_export.xlsx"

DBMS=xlsx REPLACE;

**RUN**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 5. Calculate MNC-level and country-level component auditor variables from Form AP

Calculates the following variables, used in the MNC analyses (see Appendix for definitions):

N Variable Name Label in Paper

1. total\_component\_perc Component Auditor Hours-%

2. component (used to calculate Any Coverage)

This step also generates a dataset of individual component auditors, identified at the country level.

This dataset is used to match individual component auditors to Orbis subsidiary observations in STEP 12

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** F1;

set Caudit.FormAP;

\* Delete observations for non-issuers (employee benefit plans and investment portfolios) and Non-U.S. Companies;

IF Audit\_Report\_Type ^= "Issuer, other than Employee Benefit Plan or Investment Company" THEN DELETE;

\* IF firm\_issuing\_country ne "United States" THEN DELETE; \* Note this will be handled as part of the merge with compustat

\*Create Variables;

\* Create/adjust Category Variables;

fye = datepart(Fiscal\_Period\_End\_Date);

format fye mmddyy10.;

year = year(fye);

FORMAT signed mmddyy10.;

signed = datepart(signed\_date);

DROP signed\_date;

RENAME signed = signed\_date;

rename Issuer\_Cik = fkey;

ardate = datepart(audit\_report\_date);

apfiledate = datepart(filing\_date);

apdelay = apfiledate - ardate;

if Amendment\_Other\_Firms = "TRUE" then amend = **1**; else amend = **0**;

**run**;

\* Keep only initial Form AP filing, unless the component auditor data was amended (Amendment\_Other\_Firms = "TRUE");

**PROC** **SORT** DATA = F1; by fkey year DESCENDING amend signed\_date; **run**;

**PROC** **SORT** DATA = F1 nodupkey; by fkey year; **run**;

\* Create variables for country code and calculate affiliates;

**DATA** F2;

SET F1;

\* Calculate Dummy Variables for number of NON-US components;

if cmp\_name\_1 ne ' ' and cmp\_country\_1 ne "United States" then cmp\_1 = **1**; else cmp\_1 = **0**;

if cmp\_name\_2 ne ' ' and cmp\_country\_2 ne "United States" then cmp\_2 = **1**; else cmp\_2 = **0**;

if cmp\_name\_3 ne ' ' and cmp\_country\_3 ne "United States" then cmp\_3 = **1**; else cmp\_3 = **0**;

if cmp\_name\_4 ne ' ' and cmp\_country\_4 ne "United States" then cmp\_4 = **1**; else cmp\_4 = **0**;

if cmp\_name\_5 ne ' ' and cmp\_country\_5 ne "United States" then cmp\_5 = **1**; else cmp\_5 = **0**;

if cmp\_name\_6 ne ' ' and cmp\_country\_6 ne "United States" then cmp\_6 = **1**; else cmp\_6 = **0**;

if cmp\_name\_7 ne ' ' and cmp\_country\_7 ne "United States" then cmp\_7 = **1**; else cmp\_7 = **0**;

if cmp\_name\_8 ne ' ' and cmp\_country\_8 ne "United States" then cmp\_8 = **1**; else cmp\_8 = **0**;

\* Create total number of NON\_US component auditors and dummy for any level of component usage;

total\_component = sum(of cmp\_1-cmp\_8, Number\_of\_Participants);

if total\_component = **.** then total\_component = **0**;

if total\_component > **0** then component = **1**; else component = **0**;

\* Loop through component auditors to assign a country code (for matching with subsidiary observations in STEP 12);

array coun(**8**) cmp\_country\_1 cmp\_country\_2 cmp\_country\_3 cmp\_country\_4 cmp\_country\_5 cmp\_country\_6 cmp\_country\_7 cmp\_country\_8;

ARRAY countryid(**8**) cmp\_countryid\_1 cmp\_countryid\_2 cmp\_countryid\_3 cmp\_countryid\_4 cmp\_countryid\_5 cmp\_countryid\_6 cmp\_countryid\_7 cmp\_countryid\_8;

DO indexid = **1** to **8**;

IF coun(indexid) = "Algeria" THEN countryid(indexid) = **3**;

ELSE IF coun(indexid) = "Argentina" THEN countryid(indexid) = **5**;

ELSE IF coun(indexid) = "Armenia" THEN countryid(indexid) = **6**;

ELSE IF coun(indexid) = "Australia" THEN countryid(indexid) = **121**;

ELSE IF coun(indexid) = "Austria" THEN countryid(indexid) = **122**;

ELSE IF coun(indexid) = "Bahamas" THEN countryid(indexid) = **146**;

ELSE IF coun(indexid) = "Bahrain" THEN countryid(indexid) = **148**;

ELSE IF coun(indexid) = "Barbados" THEN countryid(indexid) = **150**;

ELSE IF coun(indexid) = "Belarus" THEN countryid(indexid) = **9**;

ELSE IF coun(indexid) = "Belgium" THEN countryid(indexid) = **123**;

ELSE IF coun(indexid) = "Bermuda" THEN countryid(indexid) = **154**;

ELSE IF coun(indexid) = "Bolivia, Plurinational State of" THEN countryid(indexid) = **12**;

ELSE IF coun(indexid) = "Brazil" THEN countryid(indexid) = **108**;

ELSE IF coun(indexid) = "Bulgaria" THEN countryid(indexid) = **15**;

ELSE IF coun(indexid) = "Canada" THEN countryid(indexid) = **124**;

ELSE IF coun(indexid) = "Cayman Islands" THEN countryid(indexid) = **158**;

ELSE IF coun(indexid) = "Chile" THEN countryid(indexid) = **22**;

ELSE IF coun(indexid) = "China" THEN countryid(indexid) = **23**;

ELSE IF coun(indexid) = "Colombia" THEN countryid(indexid) = **24**;

ELSE IF coun(indexid) = "Costa Rica" THEN countryid(indexid) = **27**;

ELSE IF coun(indexid) = "Curacao" THEN countryid(indexid) = **300**;

ELSE IF coun(indexid) = "Cyprus" THEN countryid(indexid) = **168**;

ELSE IF coun(indexid) = "Czech Republic" THEN countryid(indexid) = **109**;

ELSE IF coun(indexid) = "Denmark" THEN countryid(indexid) = **125**;

ELSE IF coun(indexid) = "Dominican Republic" THEN countryid(indexid) = **32**;

ELSE IF coun(indexid) = "Egypt" THEN countryid(indexid) = **34**;

ELSE IF coun(indexid) = "El Salvador" THEN countryid(indexid) = **35**;

ELSE IF coun(indexid) = "Finland" THEN countryid(indexid) = **126**;

ELSE IF coun(indexid) = "France" THEN countryid(indexid) = **127**;

ELSE IF coun(indexid) = "Germany" THEN countryid(indexid) = **128**;

ELSE IF coun(indexid) = "Ghana" THEN countryid(indexid) = **41**;

ELSE IF coun(indexid) = "Greece" THEN countryid(indexid) = **129**;

ELSE IF coun(indexid) = "Honduras" THEN countryid(indexid) = **48**;

ELSE IF coun(indexid) = "Hong Kong" THEN countryid(indexid) = **130**;

ELSE IF coun(indexid) = "Hungary" THEN countryid(indexid) = **110**;

ELSE IF coun(indexid) = "Iceland" THEN countryid(indexid) = **132**;

ELSE IF coun(indexid) = "India" THEN countryid(indexid) = **49**;

ELSE IF coun(indexid) = "Indonesia" THEN countryid(indexid) = **50**;

ELSE IF coun(indexid) = "Ireland" THEN countryid(indexid) = **134**;

ELSE IF coun(indexid) = "Israel" THEN countryid(indexid) = **136**;

ELSE IF coun(indexid) = "Italy" THEN countryid(indexid) = **139**;

ELSE IF coun(indexid) = "Jamaica" THEN countryid(indexid) = **53**;

ELSE IF coun(indexid) = "Japan" THEN countryid(indexid) = **141**;

ELSE IF coun(indexid) = "Jersey" THEN countryid(indexid) = **401**;

ELSE IF coun(indexid) = "Kazakhstan" THEN countryid(indexid) = **55**;

ELSE IF coun(indexid) = "Korea, Republic of" THEN countryid(indexid) = **111**;

ELSE IF coun(indexid) = "Latvia" THEN countryid(indexid) = **59**;

ELSE IF coun(indexid) = "Lithuania" THEN countryid(indexid) = **115**;

ELSE IF coun(indexid) = "Luxembourg" THEN countryid(indexid) = **143**;

ELSE IF coun(indexid) = "Macao" THEN countryid(indexid) = **180**;

ELSE IF coun(indexid) = "Malawi" THEN countryid(indexid) = **65**;

ELSE IF coun(indexid) = "Malaysia" THEN countryid(indexid) = **145**;

ELSE IF coun(indexid) = "Malta" THEN countryid(indexid) = **181**;

ELSE IF coun(indexid) = "Mexico" THEN countryid(indexid) = **69**;

ELSE IF coun(indexid) = "Mongolia" THEN countryid(indexid) = **71**;

ELSE IF coun(indexid) = "Mozambique" THEN countryid(indexid) = **73**;

ELSE IF coun(indexid) = "Namibia" THEN countryid(indexid) = **187**;

ELSE IF coun(indexid) = "Netherlands" THEN countryid(indexid) = **147**;

ELSE IF coun(indexid) = "New Zealand" THEN countryid(indexid) = **149**;

ELSE IF coun(indexid) = "Nicaragua" THEN countryid(indexid) = **75**;

ELSE IF coun(indexid) = "Nigeria" THEN countryid(indexid) = **77**;

ELSE IF coun(indexid) = "Norway" THEN countryid(indexid) = **151**;

ELSE IF coun(indexid) = "Pakistan" THEN countryid(indexid) = **78**;

ELSE IF coun(indexid) = "Panama" THEN countryid(indexid) = **193**;

ELSE IF coun(indexid) = "Paraguay" THEN countryid(indexid) = **79**;

ELSE IF coun(indexid) = "Peru" THEN countryid(indexid) = **80**;

ELSE IF coun(indexid) = "Philippines" THEN countryid(indexid) = **81**;

ELSE IF coun(indexid) = "Poland" THEN countryid(indexid) = **82**;

ELSE IF coun(indexid) = "Portugal" THEN countryid(indexid) = **153**;

ELSE IF coun(indexid) = "Romania" THEN countryid(indexid) = **83**;

ELSE IF coun(indexid) = "Russian Federation" THEN countryid(indexid) = **84**;

ELSE IF coun(indexid) = "Serbia" THEN countryid(indexid) = **89**;

ELSE IF coun(indexid) = "Singapore" THEN countryid(indexid) = **155**;

ELSE IF coun(indexid) = "Slovakia" THEN countryid(indexid) = **91**;

ELSE IF coun(indexid) = "Slovenia" THEN countryid(indexid) = **117**;

ELSE IF coun(indexid) = "South Africa" THEN countryid(indexid) = **92**;

ELSE IF coun(indexid) = "Spain" THEN countryid(indexid) = **157**;

ELSE IF coun(indexid) = "Sri Lanka" THEN countryid(indexid) = **93**;

ELSE IF coun(indexid) = "Sweden" THEN countryid(indexid) = **159**;

ELSE IF coun(indexid) = "Switzerland" THEN countryid(indexid) = **161**;

ELSE IF coun(indexid) = "Taiwan, Province of China" THEN countryid(indexid) = **163**;

ELSE IF coun(indexid) = "Tanzania, United Republic of" THEN countryid(indexid) = **98**;

ELSE IF coun(indexid) = "Thailand" THEN countryid(indexid) = **99**;

ELSE IF coun(indexid) = "Trinidad and Tobago" THEN countryid(indexid) = **102**;

ELSE IF coun(indexid) = "Turkey" THEN countryid(indexid) = **104**;

ELSE IF coun(indexid) = "Ukraine" THEN countryid(indexid) = **106**;

ELSE IF coun(indexid) = "United Arab Emirates" THEN countryid(indexid) = **208**;

ELSE IF coun(indexid) = "United Kingdom" THEN countryid(indexid) = **165**;

ELSE IF coun(indexid) = "United States" THEN countryid(indexid) = **167**;

ELSE IF coun(indexid) = "Uruguay" THEN countryid(indexid) = **107**;

ELSE IF coun(indexid) = "Viet Nam" THEN countryid(indexid) = **133**;

ELSE IF coun(indexid) = "Zambia" THEN countryid(indexid) = **137**;

ELSE IF coun(indexid) = "Zimbabwe" THEN countryid(indexid) = **14**;

ELSE IF coun(indexid) = "Guatemala" THEN countryid(indexid) = **43**;

ELSE IF coun(indexid) = "Morocco" THEN countryid(indexid) = **72**;

ELSE IF coun(indexid) = "Gibraltar" THEN countryid(indexid) = **222**;

END;

\* Set percentage of total audit hours to zero (or missing) if the component auditor is in the U.S.;

if cmp\_countryid\_1 = **167** then cmp\_perc\_1 = **.**; if cmp\_countryid\_2 = **167** then cmp\_perc\_2 = **.**; if cmp\_countryid\_3 = **167** then cmp\_perc\_3 = **.**;

if cmp\_countryid\_4 = **167** then cmp\_perc\_4 = **.**; if cmp\_countryid\_5 = **167** then cmp\_perc\_5 = **.**; if cmp\_countryid\_6 = **167** then cmp\_perc\_6 = **.**;

if cmp\_countryid\_7 = **167** then cmp\_perc\_7 = **.**; if cmp\_countryid\_8 = **167** then cmp\_perc\_8 = **.**;

**run**;

\* Create midpoint values for Component % Ranges;

**DATA** F3;

SET F2;

ARRAY range(**8**) cmp\_range\_1 cmp\_range\_2 cmp\_range\_3 cmp\_range\_4 cmp\_range\_5 cmp\_range\_6 cmp\_range\_7 cmp\_range\_8;

ARRAY country(**8**) cmp\_country\_1 cmp\_country\_2 cmp\_country\_3 cmp\_country\_4 cmp\_country\_5 cmp\_country\_6 cmp\_country\_7 cmp\_country\_8;

ARRAY midpoint(**8**) cmp\_mdpt\_1 cmp\_mdpt\_2 cmp\_mdpt\_3 cmp\_mdpt\_4 cmp\_mdpt\_5 cmp\_mdpt\_6 cmp\_mdpt\_7 cmp\_mdpt\_8;

DO indexmdpt = **1** to **8**;

IF range(indexmdpt) = "Less than 5%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **2.5**; \*Nested if to calculate midpoint of % range;

ELSE IF range(indexmdpt) = "5% to less than 10%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **7.5**;

ELSE IF range(indexmdpt) = "10% to less than 20%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **15**;

ELSE IF range(indexmdpt) = "20% to less than 30%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **25**;

ELSE IF range(indexmdpt) = "30% to less than 40%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **35**;

ELSE IF range(indexmdpt) = "40% to less than 50%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **45**;

ELSE IF range(indexmdpt) = "50% to less than 60%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **55**;

ELSE IF range(indexmdpt) = "60% to less than 70%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **65**;

ELSE IF range(indexmdpt) = "70% to less than 80%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **75**;

ELSE IF range(indexmdpt) = "80% to less than 90%" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **85**;

ELSE IF range(indexmdpt) = "90% or more" and country(indexmdpt) ne "United States" then midpoint(indexmdpt)= **95**;

ELSE midpoint(indexmdpt)= **.**;

END;

\*Calculate Midpoint value for all component auditors <5% ("participant");

IF Participant\_range = "Less than 5%" then participant\_mdpt= **2.5**; \*Nested if to calculate midpoint of % range;

ELSE IF Participant\_range = "5% to less than 10%" then participant\_mdpt= **7.5**;

ELSE IF Participant\_range = "10% to less than 20%" then participant\_mdpt= **15**;

ELSE IF Participant\_range = "20% to less than 30%" then participant\_mdpt= **25**;

ELSE IF Participant\_range = "30% to less than 40%" then participant\_mdpt= **35**;

ELSE IF Participant\_range = "40% to less than 50%" then participant\_mdpt= **45**;

ELSE IF Participant\_range = "50% to less than 60%" then participant\_mdpt= **55**;

ELSE IF Participant\_range = "60% to less than 70%" then participant\_mdpt= **65**;

ELSE IF Participant\_range = "70% to less than 80%" then participant\_mdpt= **75**;

ELSE IF Participant\_range = "80% to less than 90%" then participant\_mdpt= **85**;

ELSE IF Participant\_range = "90% or more" then participant\_mdpt= **95**;

ELSE participant\_mdpt= **.**;

\* Create individual component percentage variables;

if cmp\_perc\_1 ne **.** then cmp\_hrs1 = cmp\_perc\_1;else cmp\_hrs1 = cmp\_mdpt\_1;

if cmp\_perc\_2 ne **.** then cmp\_hrs2 = cmp\_perc\_2;else cmp\_hrs2 = cmp\_mdpt\_2;

if cmp\_perc\_3 ne **.** then cmp\_hrs3 = cmp\_perc\_3;else cmp\_hrs3 = cmp\_mdpt\_3;

if cmp\_perc\_4 ne **.** then cmp\_hrs4 = cmp\_perc\_4;else cmp\_hrs4 = cmp\_mdpt\_4;

if cmp\_perc\_5 ne **.** then cmp\_hrs5 = cmp\_perc\_5;else cmp\_hrs5 = cmp\_mdpt\_5;

if cmp\_perc\_6 ne **.** then cmp\_hrs6 = cmp\_perc\_6;else cmp\_hrs6 = cmp\_mdpt\_6;

if cmp\_perc\_7 ne **.** then cmp\_hrs7 = cmp\_perc\_7;else cmp\_hrs7 = cmp\_mdpt\_7;

cmp\_hrs8 = cmp\_mdpt\_8; \*cmp\_perc\_8 is character. There are no observations with a non-missing value, so just use midpoint;

\* Create variable that sums total component auditor %;

total\_component\_perc = sum(of cmp\_mdpt\_1-cmp\_mdpt\_8, of cmp\_perc\_1-cmp\_perc\_8, participant\_mdpt, participant\_percentage);

if total\_component\_perc = **.** then total\_component\_perc = **0**;

**RUN**;

\* Winsorize component auditor data;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Winsorize macro

Can winsorize or trim at specified percentiles;

dsetout = leave blank to overwrite dsetin

byvar = none for no byvar (trims/winsorizes pooled sample)

type = delete/winsor

ex: %winsor(dsetin=mydata, dsetout=mydata2, byvar=year, vars=assets earnings, pctl=0 98);

winsorizes by year at 98%, puts resulting dataset into mydata2

%winsor(dsetin=mydata, vars=assets earnings, type=delete);

trims pooled sample at 1% and 99%, puts resulting dataset back into mydata

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**%macro** winsor(dsetin=F3, dsetout = , byvar=none, vars=

total\_component\_perc,

type=winsor, pctl=**1** **99**);

\*nothing gets changed below\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

%if &dsetout = %then %let dsetout = &dsetin;

%let varL=;

%let varH=;

%let xn=1;

%do %until ( %scan(&vars,&xn)= );

%let token = %scan(&vars,&xn);

%let varL = &varL &token.L;

%let varH = &varH &token.H;

%let xn=%EVAL(&xn + 1);

%end;

%let xn=%eval(&xn-1);

data xtemp;

set &dsetin;

run;

%if &byvar = none %then %do;

data xtemp;

set xtemp;

xbyvar = **1**;

run;

%let byvar = xbyvar;

%end;

proc sort data = xtemp; by &byvar; run;

proc univariate data = xtemp noprint;

by &byvar;

var &vars;

output out = xtemp\_pctl PCTLPTS = &pctl PCTLPRE = &vars PCTLNAME = L H;

run;

data &dsetout;

merge xtemp xtemp\_pctl;

by &byvar;

array trimvars{&xn} &vars;

array trimvarl{&xn} &varL;

array trimvarh{&xn} &varH;

do xi = **1** to dim(trimvars);

%if &type = winsor %then %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then trimvars{xi} = trimvarl{xi};

if (trimvars{xi} > trimvarh{xi}) then trimvars{xi} = trimvarh{xi};

end;

%end;

%else %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then delete;

if (trimvars{xi} > trimvarh{xi}) then delete;

end;

%end;

end;

drop &varL &varH xbyvar xi;

run;

**%mend** winsor;

%***winsor***

**run**; \*executes the macro;

**proc** **sort** data=F3 nodupkey; by fkey year; **run**;

\* Delete observations before the Form AP Component Auditor requirement (audit reports issued on or after 6/30/2017)

or observations with fiscal year ends before March 31, 2017 or with abnormally late Form AP filings (greater than 60 days delay)

Save dataset to library;

**DATA** Caudit.F1;

SET F3;

format ardate mmddyy10.;

if fkey = **.** then delete; \* drops 2 observations;

IF ardate < **'30JUN2017'd** THEN DELETE;

\* delete observations for audit reports issued abnormally late (e.g., multiple years issued in the same report);

IF fye < **'31MAR2017'd** THEN DELETE;

if apdelay > **60** & amend = **0** THEN DELETE;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Generate dataset of Individually identified component auditors (those contributing > 5% of total audit hours)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data=F3; by fkey year firm\_name firm\_id firm\_country; **run**;

\* Generate Dataset that uses Firm-Year-Component Auditor as the observation level;

**proc** **transpose** data=F3 out=Tr1 (rename=(\_NAME\_ = transpose COL1 = cmp\_countryid));

var cmp\_countryid\_1 cmp\_countryid\_2 cmp\_countryid\_3 cmp\_countryid\_4 cmp\_countryid\_5 cmp\_countryid\_6 cmp\_countryid\_7 cmp\_countryid\_8;

by fkey year firm\_name firm\_id firm\_country;

**run**;

\* Transpose Component Auditor Hours for Each Observation;

**proc** **transpose** data=F3 out=Tr2 (rename=(\_NAME\_ = transpose COL1 = cmp\_hrs));

var cmp\_hrs1 cmp\_hrs2 cmp\_hrs3 cmp\_hrs4 cmp\_hrs5 cmp\_hrs6 cmp\_hrs7 cmp\_hrs8;

by fkey year firm\_name firm\_id firm\_country;

**run**;

\* Transpose Component Auditor Country Name;

**proc** **transpose** data=F3 out=Tr2a (rename=(\_NAME\_ = transpose COL1 = cmp\_country));

var cmp\_country\_1 cmp\_country\_2 cmp\_country\_3 cmp\_country\_4 cmp\_country\_5 cmp\_country\_6 cmp\_country\_7 cmp\_country\_8;

by fkey year firm\_name firm\_id firm\_country;

**run**;

\* Transpose Component Auditor ID;

**proc** **transpose** data=F3 out=Tr2b (rename=(\_NAME\_ = transpose COL1 = cmp\_firmid));

var cmp\_firm\_id\_1 cmp\_firm\_id\_2 cmp\_firm\_id\_3 cmp\_firm\_id\_4 cmp\_firm\_id\_5 cmp\_firm\_id\_6 cmp\_firm\_id\_7 cmp\_firm\_id\_8;

by fkey year firm\_name firm\_id firm\_country;

**run**;

\* Transpose Component Auditor Name;

**proc** **transpose** data=F3 out=Tr2c (rename=(\_NAME\_ = transpose COL1 = cmp\_name));

var cmp\_name\_1 cmp\_name\_2 cmp\_name\_3 cmp\_name\_4 cmp\_name\_5 cmp\_name\_6 cmp\_name\_7 cmp\_name\_8;

by fkey year firm\_name firm\_id firm\_country;

**run**;

\* Transpose Component Auditor City;

**proc** **transpose** data=F3 out=Tr2d (rename=(\_NAME\_ = transpose COL1 = cmp\_city));

var cmp\_city\_1 cmp\_city\_2 cmp\_city\_3 cmp\_city\_4 cmp\_city\_5 cmp\_city\_6 cmp\_city\_7 cmp\_city\_8;

by fkey year firm\_name firm\_id firm\_country;

**run**;

**data** Tr3;

merge tr1 tr2 tr2a tr2b tr2c tr2d;

**run**;

\* Delete observations with missing component auditors;

**data** Tr3a;

set Tr3;

if cmp\_countryid ne **.**;

**run**;

\* Check for missing countries - dataset should have 0 observations;

**data** ch;

set Tr3;

if cmp\_countryid ne **.** then delete;

**run**;

\* Delete U.S. component auditors;

**data** Tr4;

set Tr3a;

if cmp\_countryid = **167** then delete;

cmp = **1**;

**run**;

\* There are several observations with duplicates (more than one component auditor within firm-year-country).

We retain the component auditor within each country that has the highest % of hours. This does not affect our analyses in any way;

**proc** **sort** data=Tr4; by fkey year cmp\_countryid descending cmp\_hrs; **run**;

**proc** **sort** data=Tr4 out=Tr5 nodupkey; by fkey year cmp\_countryid; **run**;

\* Save dataset for use in STEP 12;

**data** Caudit.cmpcountries;

set Tr5;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 6. Import Audit Analytics Data

Files were downloaded directly from Audit Analytics and saved as Excel files.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Audit Opinion Data;

**PROC** **IMPORT** OUT= work.auditopin

DATAFILE= "FILEPATH\Opin.xlsx"

DBMS=xlsx REPLACE;

**RUN**;

\* Financial Statement Restatements Data;

**PROC** **IMPORT** OUT= Caudit.restatements

DATAFILE= "FILEPATH\restatement-data.xlsx"

DBMS=xlsx REPLACE;

**RUN**;

**data** opin ;

set Work.auditopin(rename = (CIK\_Code = fkey year\_ended\_date = datadate Year\_Ended\_\_Ideal\_ = year auditor\_key = aud\_num

Auditor\_City = aud\_city Auditor\_State\_Code = aud\_state SIC\_Code = sic VAR55 = audit\_fees

VAR56 = nonaudit\_fees VAR57 = total\_fees Auditor = auditor\_name source\_date = op\_file\_date));

if year < **2010** then delete;

format datadate mmddyy10.;

format opinion\_date mmddyy10.;

opinion\_date = signature\_date;\* - ('01jan1960'd - '30dec1899'd);

if Auditor\_State\_Region = "Canada" or Auditor\_State\_Region = "Foreign" then delete;

keep fkey datadate year aud\_num aud\_city aud\_state sic audit\_fees nonaudit\_fees total\_fees auditor\_name

going\_concern opinion\_date op\_file\_date filer\_status;

**run**;

\* Delete duplicates - delete obs with smaller fees when one firm has > 1 auditor in a year;

**proc** **sort** data=opin; by fkey year descending audit\_fees; **run**;

**proc** **sort** data=opin nodupkey; by fkey year; **run**;

\* Save AA dataset to library;

**data** Caudit.AA\_data;

set opin;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 7. Calculate Audit Analytics Variables

Calculates the following variables (see Appendix for definitions):

N Variable Name Label in Paper

1. MNC\_accelerated MNC Accelerated

2. MNC\_industry\_expert MNC Industry Expert

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Calculate control variables;

**proc** **sort** data=Caudit.AA\_data out=a1 nodupkey;by fkey year;**run**;

**data** A2;

set a1;

\* Define MNC\_accelerated;

if filer\_status = "Large Accelerated Filer" or filer\_status = "Accelerated Filer" then MNC\_accelerated = **1**; else MNC\_accelerated = **0**;

\* Calculate SIC 2-digit industry codes and redefine aud\_city for industry expertise steps below);

sic2 = int(sic / **100**);

aud\_city = upcase(aud\_city);

**run**;

\* Merge dataset with hand-collected CBSA Codes basedon auditor city and state to calculate City-level Variables;

**proc** **sort** data=A2; by aud\_city aud\_state;**run**;

**proc** **sort** data=Caudit.cbsa out=cbsa;by aud\_city aud\_state;**run**;

**data** A3;

merge A2 (in=xx) cbsa (in=yy);

by aud\_city aud\_state;

if xx;

**run**;

**proc** **sort** data=A3;by audcbsacode;**run**;

\* Check for missing data;

**proc** **means** data=A3 n nmiss;var audcbsacode;**run**;

\*Sum Client Fees by Auditor City and define Industry Expert ;

**proc** **sort** data=A3;by year audcbsacode sic2;

**proc** **means** data=A3 sum noprint; \* sum audit fees by year, city, and industry;

by year audcbsacode sic2;

var audit\_fees;

output out=tot\_fees\_cbsa sum=fees\_cbsa;

**run**;

**data** A3a (rename = (\_freq\_ = tot\_num\_cbsa));

set tot\_fees\_cbsa; \* rename frequency to total number of clients per year-city;

drop \_type\_;

**run**;

**proc** **sort** data=A3;by year audcbsacode sic2 aud\_num; **run**;

**proc** **means** data=A3 sum noprint; \* sum audit fees by year, city, industry and auditor ;

by year audcbsacode sic2 aud\_num;

var audit\_fees;

output out=tot\_fees\_cbsa\_aud sum=fees\_cbsa\_aud;

**run**;

**data** A3b (rename = (\_freq\_ = tot\_num\_cbsa\_aud));

set tot\_fees\_cbsa\_aud; \* rename frequency to total number of clients per auditor-year ;

drop \_type\_;

**run**;

**data** A4;

merge A3a A3b;

by year audcbsacode sic2;

**run**;

\* calc and share of industry fees by auditor, and assign industry expert to those with > 50%;

**data** A5;

set A4;

if fees\_cbsa ^= **0** then shr\_industryfees\_cbsa = fees\_cbsa\_aud / fees\_cbsa;

else shr\_industryfees\_cbsa = **.**;

if shr\_industryfees\_cbsa ne **.** and shr\_industryfees\_cbsa > **0.5** then MNC\_industry\_expert = **1**; else MNC\_industry\_expert = **0**;

**run**;

**proc** **means** data=A5 n nmiss mean median min max;**run**;

\*Merge back with main dataset;

**proc** **sort** data=A5;by year audcbsacode sic2 aud\_num;**run**;

**proc** **sort** data=A3;by year audcbsacode sic2 aud\_num;**run**;

\* Save dataset to library;

**data** Caudit.A0;

merge A3 (in=xx) A5 (in=yy);

by year audcbsacode sic2 aud\_num;

if yy;

**run**;

**proc** **means** data=Caudit.A0;**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 8. Merge MNC-level datasets and calculate additional controls

Calculates the following variables used in the analyses:

N Variable Name Label in Paper

1. restate\_a MNC Restate

2. lnarc MNC ARC

3. MNC\_big4 MNC Big 4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Add numerical CIK code to Compustat Dataset;

**proc** **sort** data=Caudit.C1 out=C1 nodupkey;by cik year;**run**;

**data** C2;

set C1;

cik2 = cik\***1**;

**run**;

\* Merge in MNC Restatement variables;

**data** res;

set Caudit.restatements;

rename CIK\_Code = cik;

rename Restated\_Period\_Begin = resbegin;

rename Restated\_Period\_Ended = resend;

rename Disclosure\_Date = disdate;

restate = **1**;

**run**;

\* Merge Compustat Data with restatement data based on begin/end restatement date relative to datadate (take any restatement within 1-year prior to datadate);

**proc** **sql**;

create table r1 as

select a.\*, b.restate as restate\_a, b.resbegin, b.resend, b.disdate, b.disclosure

from C2 as a left join res as b

on (a.cik2 = b.cik) and (b.resbegin < a.datadate) and (b.resend >= a.datadate-**10**);

**quit**;

\* Keep only most recent restatement data for each observation;

**proc** **sort** data=r1; by cik2 year descending restate\_a descending disdate; **run**;

**data** r1;

set r1;

by cik2 year descending restate\_a;

if first.year;

if restate\_a = **.** then restate\_a = **0**;

**run**;

**proc** **means** data=r1;**run**;

\*Merge with Audit Analytics;

**proc** **sort** data=r1 nodupkey;by cik year;**run**;

**proc** **sort** data=Caudit.A0 out=A1 nodupkey;by fkey year;**run**;

**data** a2;

set A1 (rename = (fkey = cik2));

drop sic sic2 datadate;

**run**;

**data** m1;

merge r1 (in=xx) a2 (in=yy);

by cik2 year;

if xx;

if MNC\_industry\_expert = **.** then MNC\_industry\_expert = **0**;

**run**;

**proc** **means** data=m1;**run**;

**proc** **sort** data=m1;by gvkey year;**run**;

\*Merge with Form AP and delete observations with missing variables;

**proc** **sort** data=m1 nodupkey;by cik2 year; **run**;

**proc** **sort** data=Caudit.F1 out=F2 nodupkey;by fkey year; **run**;

**data** F2a;

set F2;

format fiscal\_period\_end\_date mmddyy10.;

**run**;

\* Drop observations after 12/31/2019;

**data** F2b;

set F2a;

if year > **2019** then delete;

**run**;

\* Drop observations audited by firms outside the U.S.;

**data** F3;

set F2b;

if Firm\_Country ne "United States" then delete;

**run**;

**proc** **sql**;

create table M2 as

select a.\*,b.\*

from F3 a left join M1 b

on a.fkey=b.cik2 and a.year=b.year;

**quit**;

\* Delete observations with key data missing and generate Big 4 variable based on Form AP disclosures;

**DATA** M3;

SET M2;

if at = **.** then delete;

if at <=**1** then delete;

if sale = **.** then delete;

IF firm\_id = **42** or firm\_id = **34** or firm\_id = **238** or firm\_id = **185** THEN big4 = **1**; ELSE big4 = **0**;

**run**;

\* Calculate Log of totalnumber of distinct monetary XBRL tags (ARC) and

merge data. Downloaded directly from http://www.xbrlresearch.com (credit Hoitash and Hoitash [2018]);

**data** arc;

set Caudit.arc;

format fiscal\_period\_end mmddyy10.;

lnarc = log(arc);

**run**;

**proc** **sql**;

create table M4 as

select a.\*, b.arc, b.lnarc

from M3 as a left join arc as b

on a.cik2 = b.cik and a.datadate = b.fiscal\_period\_end;

**quit**;

**proc** **sort** data=M4 nodupkey; by gvkey year; **run**;

**proc** **means** n nmiss data=m4; var cik2 arc lnarc mnc\_size datadate; **run**;

\*\* Winsorize Variables at 99%;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Winsorize macro

Can winsorize or trim at specified percentiles;

dsetout = leave blank to overwrite dsetin

byvar = none for no byvar (trims/winsorizes pooled sample)

type = delete/winsor

ex: %winsor(dsetin=mydata, dsetout=mydata2, byvar=year, vars=assets earnings, pctl=0 98);

winsorizes by year at 98%, puts resulting dataset into mydata2

%winsor(dsetin=mydata, vars=assets earnings, type=delete);

trims pooled sample at 1% and 99%, puts resulting dataset back into mydata

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**%macro** winsor(dsetin=M4, dsetout = , byvar=none, vars=

arc lnarc,

type=winsor, pctl=**1** **99**);

\*nothing gets changed below\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

%if &dsetout = %then %let dsetout = &dsetin;

%let varL=;

%let varH=;

%let xn=1;

%do %until ( %scan(&vars,&xn)= );

%let token = %scan(&vars,&xn);

%let varL = &varL &token.L;

%let varH = &varH &token.H;

%let xn=%EVAL(&xn + 1);

%end;

%let xn=%eval(&xn-1);

data xtemp;

set &dsetin;

run;

%if &byvar = none %then %do;

data xtemp;

set xtemp;

xbyvar = **1**;

run;

%let byvar = xbyvar;

%end;

proc sort data = xtemp; by &byvar; run;

proc univariate data = xtemp noprint;

by &byvar;

var &vars;

output out = xtemp\_pctl PCTLPTS = &pctl PCTLPRE = &vars PCTLNAME = L H;

run;

data &dsetout;

merge xtemp xtemp\_pctl;

by &byvar;

array trimvars{&xn} &vars;

array trimvarl{&xn} &varL;

array trimvarh{&xn} &varH;

do xi = **1** to dim(trimvars);

%if &type = winsor %then %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then trimvars{xi} = trimvarl{xi};

if (trimvars{xi} > trimvarh{xi}) then trimvars{xi} = trimvarh{xi};

end;

%end;

%else %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then delete;

if (trimvars{xi} > trimvarh{xi}) then delete;

end;

%end;

end;

drop &varL &varH xbyvar xi;

run;

**%mend** winsor;

%***winsor***

**run**; \*executes the macro;

\* Check duplicates (should be 0);

**proc** **sort** data=M4 nodupkey; by gvkey year; **run**;

\* Save to SAS library;

**data** Caudit.MNC;

set M4;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 9. Export MNC dataset to Stata

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **export**

data=Caudit.MNC

dbms=dta

outfile = 'FILEPATH\MNC' replace;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Generate Subsidiary Dataset

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 10. Import Orbis Subsidiary Data

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

/\* IMPORT Subsidiary Data from Orbis\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

DM "Output; Clear; Log; Clear";

option nolabel;

**PROC** **DATASETS** LIB=work KILL;

options helpbrowser=sas;**run**;**quit**;

\* Import subsidiary data obtained from Orbis database;

**PROC** **IMPORT** OUT= work.orbis\_det

DATAFILE= "FILEPATH\Form AP Subs Detail\_SAS.xlsx"

DBMS=xlsx REPLACE;

**RUN**;

\* Import Owner data (for ticker) obtained from Orbis database;

**PROC** **IMPORT** OUT= work.orbis\_id

DATAFILE= "FILEPATH\Form AP Identifiers\_SAS.xlsx"

DBMS=xlsx REPLACE;

**RUN**;

\* Add CIK number of parent company to each subs dataset;

**DATA** orbis\_det1;

set orbis\_det;

rename GUO\_\_\_BvD\_ID\_number = parent\_bvd\_id;

**run**;

**proc** **sql**;

create table orbis\_det2 as

select a.\*, b.cik\_complete as cik, b.country\_iso\_code as parent\_country

from orbis\_det1 as a left join orbis\_id as b

on a.parent\_bvd\_id = b.BvD\_ID\_number;

**quit**;

\* Rename Variables from Orbis and convert units from thousands USD to millions USD;

**Data** orbis\_det3;

set work.orbis\_det2;

Rename US\_SIC\_core\_code\_\_3\_digits\_ = SIC3;

fa\_tot\_sub\_2019 = Fixed\_assets\_th\_USD\_2019 / **1000**;

fa\_tot\_sub\_2018 = Fixed\_assets\_th\_USD\_2018 / **1000**;

fa\_tot\_sub\_2017 = Fixed\_assets\_th\_USD\_2017 / **1000**;

fa\_tot\_sub\_2016 = Fixed\_assets\_th\_USD\_2016 / **1000**;

fa\_tot\_sub\_2015 = Fixed\_assets\_th\_USD\_2015 / **1000**;

intang\_\_sub\_2019 = Intangible\_fixed\_assets\_th\_USD\_2 / **1000**;

intang\_\_sub\_2018 = Intangible\_fixed\_assets\_th\_USD\_1 / **1000**;

intang\_\_sub\_2017 = Intangible\_fixed\_assets\_th\_USD\_3 / **1000**;

intang\_\_sub\_2016 = Intangible\_fixed\_assets\_th\_USD\_4 / **1000**;

intang\_\_sub\_2015 = Intangible\_fixed\_assets\_th\_USD\_5 / **1000**;

ppe\_sub\_2019 = Tangible\_fixed\_assets\_th\_USD\_201 / **1000**;

ppe\_sub\_2018 = Tangible\_fixed\_assets\_th\_USD\_202 / **1000**;

ppe\_sub\_2017 = Tangible\_fixed\_assets\_th\_USD\_203 / **1000**;

ppe\_sub\_2016 = Tangible\_fixed\_assets\_th\_USD\_204 / **1000**;

ppe\_sub\_2015 = Tangible\_fixed\_assets\_th\_USD\_205 / **1000**;

fa\_oth\_sub\_2019 = Other\_fixed\_assets\_th\_USD\_2019 / **1000**;

fa\_oth\_sub\_2018 = Other\_fixed\_assets\_th\_USD\_2018 / **1000**;

fa\_oth\_sub\_2017 = Other\_fixed\_assets\_th\_USD\_2017 / **1000**;

fa\_oth\_sub\_2016 = Other\_fixed\_assets\_th\_USD\_2016 / **1000**;

fa\_oth\_sub\_2015 = Other\_fixed\_assets\_th\_USD\_2015 / **1000**;

ca\_sub\_2019 = Current\_assets\_th\_USD\_2019 / **1000**;

ca\_sub\_2018 = Current\_assets\_th\_USD\_2018 / **1000**;

ca\_sub\_2017 = Current\_assets\_th\_USD\_2017 / **1000**;

ca\_sub\_2016 = Current\_assets\_th\_USD\_2016 / **1000**;

ca\_sub\_2015 = Current\_assets\_th\_USD\_2015 / **1000**;

inv\_sub\_2019 = Stock\_th\_USD\_2019 / **1000**;

inv\_sub\_2018 = Stock\_th\_USD\_2018 / **1000**;

inv\_sub\_2017 = Stock\_th\_USD\_2017 / **1000**;

inv\_sub\_2016 = Stock\_th\_USD\_2016 / **1000**;

inv\_sub\_2015 = Stock\_th\_USD\_2015 / **1000**;

ar\_sub\_2019 = Debtors\_th\_USD\_2019 / **1000**;

ar\_sub\_2018 = Debtors\_th\_USD\_2018 / **1000**;

ar\_sub\_2017 = Debtors\_th\_USD\_2017 / **1000**;

ar\_sub\_2016 = Debtors\_th\_USD\_2016 / **1000**;

ar\_sub\_2015 = Debtors\_th\_USD\_2015 / **1000**;

oca\_sub\_2019 = Other\_current\_assets\_th\_USD\_2019 / **1000**;

oca\_sub\_2018 = Other\_current\_assets\_th\_USD\_2018 / **1000**;

oca\_sub\_2017 = Other\_current\_assets\_th\_USD\_2017 / **1000**;

oca\_sub\_2016 = Other\_current\_assets\_th\_USD\_2016 / **1000**;

oca\_sub\_2015 = Other\_current\_assets\_th\_USD\_2015 / **1000**;

cash\_sub\_2019 = Cash\_\_\_cash\_equivalent\_th\_USD\_20 / **1000**;

cash\_sub\_2018 = Cash\_\_\_cash\_equivalent\_th\_USD\_21 / **1000**;

cash\_sub\_2017 = Cash\_\_\_cash\_equivalent\_th\_USD\_22 / **1000**;

cash\_sub\_2016 = Cash\_\_\_cash\_equivalent\_th\_USD\_23 / **1000**;

cash\_sub\_2015 = Cash\_\_\_cash\_equivalent\_th\_USD\_24 / **1000**;

at\_sub\_2019 = Total\_assets\_th\_USD\_2019 / **1000**;

at\_sub\_2018 = Total\_assets\_th\_USD\_2018 / **1000**;

at\_sub\_2017 = Total\_assets\_th\_USD\_2017 / **1000**;

at\_sub\_2016 = Total\_assets\_th\_USD\_2016 / **1000**;

at\_sub\_2015 = Total\_assets\_th\_USD\_2015 / **1000**;

equity\_sub\_2019 = Shareholders\_funds\_th\_USD\_2019 / **1000**;

equity\_sub\_2018 = Shareholders\_funds\_th\_USD\_2018 / **1000**;

equity\_sub\_2017 = Shareholders\_funds\_th\_USD\_2017 / **1000**;

equity\_sub\_2016 = Shareholders\_funds\_th\_USD\_2016 / **1000**;

equity\_sub\_2015 = Shareholders\_funds\_th\_USD\_2015 / **1000**;

issued\_cap\_sub\_2019 = Capital\_th\_USD\_2019 / **1000**;

issued\_cap\_sub\_2018 = Capital\_th\_USD\_2018 / **1000**;

issued\_cap\_sub\_2017 = Capital\_th\_USD\_2017 / **1000**;

issued\_cap\_sub\_2016 = Capital\_th\_USD\_2016 / **1000**;

issued\_cap\_sub\_2015 = Capital\_th\_USD\_2015 / **1000**;

equity\_other\_sub\_2019 = Other\_shareholders\_funds\_th\_USD / **1000**;

equity\_other\_sub\_2018 = Other\_shareholders\_funds\_th\_USD1 / **1000**;

equity\_other\_sub\_2017 = Other\_shareholders\_funds\_th\_USD2 / **1000**;

equity\_other\_sub\_2016 = Other\_shareholders\_funds\_th\_USD3 / **1000**;

equity\_other\_sub\_2015 = Other\_shareholders\_funds\_th\_USD4 / **1000**;

noncurr\_lt\_sub\_2019 = Non\_current\_liabilities\_th\_USD\_2 / **1000**;

noncurr\_lt\_sub\_2018 = Non\_current\_liabilities\_th\_USD\_1 / **1000**;

noncurr\_lt\_sub\_2017 = Non\_current\_liabilities\_th\_USD\_3 / **1000**;

noncurr\_lt\_sub\_2016 = Non\_current\_liabilities\_th\_USD\_4 / **1000**;

noncurr\_lt\_sub\_2015 = Non\_current\_liabilities\_th\_USD\_5 / **1000**;

dltt\_sub\_2019 = Long\_term\_debt\_th\_USD\_2019 / **1000**;

dltt\_sub\_2018 = Long\_term\_debt\_th\_USD\_2018 / **1000**;

dltt\_sub\_2017 = Long\_term\_debt\_th\_USD\_2017 / **1000**;

dltt\_sub\_2016 = Long\_term\_debt\_th\_USD\_2016 / **1000**;

dltt\_sub\_2015 = Long\_term\_debt\_th\_USD\_2015 / **1000**;

noncurr\_lt\_other\_sub\_2019 = Other\_non\_current\_liabilities\_th / **1000**;

noncurr\_lt\_other\_sub\_2018 = Other\_non\_current\_liabilities\_t1 / **1000**;

noncurr\_lt\_other\_sub\_2017 = Other\_non\_current\_liabilities\_t2 / **1000**;

noncurr\_lt\_other\_sub\_2016 = Other\_non\_current\_liabilities\_t3 / **1000**;

noncurr\_lt\_other\_sub\_2015 = Other\_non\_current\_liabilities\_t4 / **1000**;

pvt\_sub\_2019 = Provisions\_th\_USD\_2019 / **1000**;

pvt\_sub\_2018 = Provisions\_th\_USD\_2018 / **1000**;

pvt\_sub\_2017 = Provisions\_th\_USD\_2017 / **1000**;

pvt\_sub\_2016 = Provisions\_th\_USD\_2016 / **1000**;

pvt\_sub\_2015 = Provisions\_th\_USD\_2015 / **1000**;

cl\_sub\_2019 = Current\_liabilities\_th\_USD\_2019 / **1000**;

cl\_sub\_2018 = Current\_liabilities\_th\_USD\_2018 / **1000**;

cl\_sub\_2017 = Current\_liabilities\_th\_USD\_2017 / **1000**;

cl\_sub\_2016 = Current\_liabilities\_th\_USD\_2016 / **1000**;

cl\_sub\_2015 = Current\_liabilities\_th\_USD\_2015 / **1000**;

std\_sub\_2019 = Loans\_th\_USD\_2019 / **1000**;

std\_sub\_2018 = Loans\_th\_USD\_2018 / **1000**;

std\_sub\_2017 = Loans\_th\_USD\_2017 / **1000**;

std\_sub\_2016 = Loans\_th\_USD\_2016 / **1000**;

std\_sub\_2015 = Loans\_th\_USD\_2015 / **1000**;

ap\_sub\_2019 = Creditors\_th\_USD\_2019 / **1000**;

ap\_sub\_2018 = Creditors\_th\_USD\_2018 / **1000**;

ap\_sub\_2017 = Creditors\_th\_USD\_2017 / **1000**;

ap\_sub\_2016 = Creditors\_th\_USD\_2016 / **1000**;

ap\_sub\_2015 = Creditors\_th\_USD\_2015 / **1000**;

other\_cl\_sub\_2019 = Other\_current\_liabilities\_th\_USD / **1000**;

other\_cl\_sub\_2018 = Other\_current\_liabilities\_th\_US1 / **1000**;

other\_cl\_sub\_2017 = Other\_current\_liabilities\_th\_US2 / **1000**;

other\_cl\_sub\_2016 = Other\_current\_liabilities\_th\_US3 / **1000**;

other\_cl\_sub\_2015 = Other\_current\_liabilities\_th\_US4 / **1000**;

lse\_sub\_2019 = Total\_shareh\_funds\_\_\_liab\_th\_USD / **1000**;

lse\_sub\_2018 = Total\_shareh\_funds\_\_\_liab\_th\_US1 / **1000**;

lse\_sub\_2017 = Total\_shareh\_funds\_\_\_liab\_th\_US2 / **1000**;

lse\_sub\_2016 = Total\_shareh\_funds\_\_\_liab\_th\_US3 / **1000**;

lse\_sub\_2015 = Total\_shareh\_funds\_\_\_liab\_th\_US4 / **1000**;

wcap\_sub\_2019 = Working\_capital\_th\_USD\_2019 / **1000**;

wcap\_sub\_2018 = Working\_capital\_th\_USD\_2018 / **1000**;

wcap\_sub\_2017 = Working\_capital\_th\_USD\_2017 / **1000**;

wcap\_sub\_2016 = Working\_capital\_th\_USD\_2016 / **1000**;

wcap\_sub\_2015 = Working\_capital\_th\_USD\_2015 / **1000**;

net\_ca\_sub\_2019 = Net\_current\_assets\_th\_USD\_2019 / **1000**;

net\_ca\_sub\_2018 = Net\_current\_assets\_th\_USD\_2018 / **1000**;

net\_ca\_sub\_2017 = Net\_current\_assets\_th\_USD\_2017 / **1000**;

net\_ca\_sub\_2016 = Net\_current\_assets\_th\_USD\_2016 / **1000**;

net\_ca\_sub\_2015 = Net\_current\_assets\_th\_USD\_2015 / **1000**;

enterprise\_value\_sub\_2019 = Enterprise\_value\_th\_USD\_2019 / **1000**;

enterprise\_value\_sub\_2018 = Enterprise\_value\_th\_USD\_2018 / **1000**;

enterprise\_value\_sub\_2017 = Enterprise\_value\_th\_USD\_2017 / **1000**;

enterprise\_value\_sub\_2016 = Enterprise\_value\_th\_USD\_2016 / **1000**;

enterprise\_value\_sub\_2015 = Enterprise\_value\_th\_USD\_2015 / **1000**;

numemp\_sub\_2019 = Number\_of\_employees\_2019 / **1000**;

numemp\_sub\_2018 = Number\_of\_employees\_2018 / **1000**;

numemp\_sub\_2017 = Number\_of\_employees\_2017 / **1000**;

numemp\_sub\_2016 = Number\_of\_employees\_2016 / **1000**;

numemp\_sub\_2015 = Number\_of\_employees\_2015 / **1000**;

oprev\_sub\_2019 = Operating\_revenue\_\_Turnover\_\_th / **1000**;

oprev\_sub\_2018 = Operating\_revenue\_\_Turnover\_\_th1 / **1000**;

oprev\_sub\_2017 = Operating\_revenue\_\_Turnover\_\_th2 / **1000**;

oprev\_sub\_2016 = Operating\_revenue\_\_Turnover\_\_th3 / **1000**;

oprev\_sub\_2015 = Operating\_revenue\_\_Turnover\_\_th4 / **1000**;

sale\_sub\_2019 = Sales\_th\_USD\_2019 / **1000**;

sale\_sub\_2018 = Sales\_th\_USD\_2018 / **1000**;

sale\_sub\_2017 = Sales\_th\_USD\_2017 / **1000**;

sale\_sub\_2016 = Sales\_th\_USD\_2016 / **1000**;

sale\_sub\_2015 = Sales\_th\_USD\_2015 / **1000**;

cogs\_sub\_2019 = Costs\_of\_goods\_sold\_th\_USD\_2019 / **1000**;

cogs\_sub\_2018 = Costs\_of\_goods\_sold\_th\_USD\_2018 / **1000**;

cogs\_sub\_2017 = Costs\_of\_goods\_sold\_th\_USD\_2017 / **1000**;

cogs\_sub\_2016 = Costs\_of\_goods\_sold\_th\_USD\_2016 / **1000**;

cogs\_sub\_2015 = Costs\_of\_goods\_sold\_th\_USD\_2015 / **1000**;

gp\_sub\_2019 = Gross\_profit\_th\_USD\_2019 / **1000**;

gp\_sub\_2018 = Gross\_profit\_th\_USD\_2018 / **1000**;

gp\_sub\_2017 = Gross\_profit\_th\_USD\_2017 / **1000**;

gp\_sub\_2016 = Gross\_profit\_th\_USD\_2016 / **1000**;

gp\_sub\_2015 = Gross\_profit\_th\_USD\_2015 / **1000**;

xopr\_sub\_2019 = Other\_operating\_expenses\_th\_USD / **1000**;

xopr\_sub\_2018 = Other\_operating\_expenses\_th\_USD1 / **1000**;

xopr\_sub\_2017 = Other\_operating\_expenses\_th\_USD2 / **1000**;

xopr\_sub\_2016 = Other\_operating\_expenses\_th\_USD3 / **1000**;

xopr\_sub\_2015 = Other\_operating\_expenses\_th\_USD4 / **1000**;

ebit\_sub\_2019 = Operating\_P\_L\_\_EBIT\_th\_USD\_2019 / **1000**;

ebit\_sub\_2018 = Operating\_P\_L\_\_EBIT\_th\_USD\_2018 / **1000**;

ebit\_sub\_2017 = Operating\_P\_L\_\_EBIT\_th\_USD\_2017 / **1000**;

ebit\_sub\_2016 = Operating\_P\_L\_\_EBIT\_th\_USD\_2016 / **1000**;

ebit\_sub\_2015 = Operating\_P\_L\_\_EBIT\_th\_USD\_2015 / **1000**;

financial\_pl\_sub\_2019 = Financial\_P\_L\_th\_USD\_2019 / **1000**;

financial\_pl\_sub\_2018 = Financial\_P\_L\_th\_USD\_2018 / **1000**;

financial\_pl\_sub\_2017 = Financial\_P\_L\_th\_USD\_2017 / **1000**;

financial\_pl\_sub\_2016 = Financial\_P\_L\_th\_USD\_2016 / **1000**;

financial\_pl\_sub\_2015 = Financial\_P\_L\_th\_USD\_2015 / **1000**;

financial\_rev\_sub\_2019 = Financial\_revenue\_th\_USD\_2019 / **1000**;

financial\_rev\_sub\_2018 = Financial\_revenue\_th\_USD\_2018 / **1000**;

financial\_rev\_sub\_2017 = Financial\_revenue\_th\_USD\_2017 / **1000**;

financial\_rev\_sub\_2016 = Financial\_revenue\_th\_USD\_2016 / **1000**;

financial\_rev\_sub\_2015 = Financial\_revenue\_th\_USD\_2015 / **1000**;

financial\_exp\_sub\_2019 = Financial\_expenses\_th\_USD\_2019 / **1000**;

financial\_exp\_sub\_2018 = Financial\_expenses\_th\_USD\_2018 / **1000**;

financial\_exp\_sub\_2017 = Financial\_expenses\_th\_USD\_2017 / **1000**;

financial\_exp\_sub\_2016 = Financial\_expenses\_th\_USD\_2016 / **1000**;

financial\_exp\_sub\_2015 = Financial\_expenses\_th\_USD\_2015 / **1000**;

ebt\_sub\_2019 = P\_L\_before\_tax\_th\_USD\_2019 / **1000**;

ebt\_sub\_2018 = P\_L\_before\_tax\_th\_USD\_2018 / **1000**;

ebt\_sub\_2017 = P\_L\_before\_tax\_th\_USD\_2017 / **1000**;

ebt\_sub\_2016 = P\_L\_before\_tax\_th\_USD\_2016 / **1000**;

ebt\_sub\_2015 = P\_L\_before\_tax\_th\_USD\_2015 / **1000**;

txt\_sub\_2019 = Taxation\_th\_USD\_2019 / **1000**;

txt\_sub\_2018 = Taxation\_th\_USD\_2018 / **1000**;

txt\_sub\_2017 = Taxation\_th\_USD\_2017 / **1000**;

txt\_sub\_2016 = Taxation\_th\_USD\_2016 / **1000**;

txt\_sub\_2015 = Taxation\_th\_USD\_2015 / **1000**;

ib\_sub\_2019 = P\_L\_after\_tax\_th\_USD\_2019 / **1000**;

ib\_sub\_2018 = P\_L\_after\_tax\_th\_USD\_2018 / **1000**;

ib\_sub\_2017 = P\_L\_after\_tax\_th\_USD\_2017 / **1000**;

ib\_sub\_2016 = P\_L\_after\_tax\_th\_USD\_2016 / **1000**;

ib\_sub\_2015 = P\_L\_after\_tax\_th\_USD\_2015 / **1000**;

xi\_sub\_2019 = Extr\_\_and\_other\_P\_L\_th\_USD\_2019 / **1000**;

xi\_sub\_2018 = Extr\_\_and\_other\_P\_L\_th\_USD\_2018 / **1000**;

xi\_sub\_2017 = Extr\_\_and\_other\_P\_L\_th\_USD\_2017 / **1000**;

xi\_sub\_2016 = Extr\_\_and\_other\_P\_L\_th\_USD\_2016 / **1000**;

xi\_sub\_2015 = Extr\_\_and\_other\_P\_L\_th\_USD\_2015 / **1000**;

xi\_rev\_sub\_2019 = Extr\_\_and\_other\_revenue\_th\_USD\_2 / **1000**;

xi\_rev\_sub\_2018 = Extr\_\_and\_other\_revenue\_th\_USD\_1 / **1000**;

xi\_rev\_sub\_2017 = Extr\_\_and\_other\_revenue\_th\_USD\_3 / **1000**;

xi\_rev\_sub\_2016 = Extr\_\_and\_other\_revenue\_th\_USD\_4 / **1000**;

xi\_rev\_sub\_2015 = Extr\_\_and\_other\_revenue\_th\_USD\_5 / **1000**;

xi\_exp\_sub\_2019 = Extr\_\_and\_other\_expenses\_th\_USD / **1000**;

xi\_exp\_sub\_2018 = Extr\_\_and\_other\_expenses\_th\_USD1 / **1000**;

xi\_exp\_sub\_2017 = Extr\_\_and\_other\_expenses\_th\_USD2 / **1000**;

xi\_exp\_sub\_2016 = Extr\_\_and\_other\_expenses\_th\_USD3 / **1000**;

xi\_exp\_sub\_2015 = Extr\_\_and\_other\_expenses\_th\_USD4 / **1000**;

ni\_sub\_2019 = P\_L\_for\_period\_\_Net\_income\_th\_US / **1000**;

ni\_sub\_2018 = P\_L\_for\_period\_\_Net\_income\_th\_U1 / **1000**;

ni\_sub\_2017 = P\_L\_for\_period\_\_Net\_income\_th\_U2 / **1000**;

ni\_sub\_2016 = P\_L\_for\_period\_\_Net\_income\_th\_U3 / **1000**;

ni\_sub\_2015 = P\_L\_for\_period\_\_Net\_income\_th\_U4 / **1000**;

exports\_sub\_2019 = Export\_revenue\_th\_USD\_2019 / **1000**;

exports\_sub\_2018 = Export\_revenue\_th\_USD\_2018 / **1000**;

exports\_sub\_2017 = Export\_revenue\_th\_USD\_2017 / **1000**;

exports\_sub\_2016 = Export\_revenue\_th\_USD\_2016 / **1000**;

exports\_sub\_2015 = Export\_revenue\_th\_USD\_2015 / **1000**;

materials\_cost\_sub\_2019 = Material\_costs\_th\_USD\_2019 / **1000**;

materials\_cost\_sub\_2018 = Material\_costs\_th\_USD\_2018 / **1000**;

materials\_cost\_sub\_2017 = Material\_costs\_th\_USD\_2017 / **1000**;

materials\_cost\_sub\_2016 = Material\_costs\_th\_USD\_2016 / **1000**;

materials\_cost\_sub\_2015 = Material\_costs\_th\_USD\_2015 / **1000**;

employees\_cost\_sub\_2019 = Costs\_of\_employees\_th\_USD\_2019 / **1000**;

employees\_cost\_sub\_2018 = Costs\_of\_employees\_th\_USD\_2018 / **1000**;

employees\_cost\_sub\_2017 = Costs\_of\_employees\_th\_USD\_2017 / **1000**;

employees\_cost\_sub\_2016 = Costs\_of\_employees\_th\_USD\_2016 / **1000**;

employees\_cost\_sub\_2015 = Costs\_of\_employees\_th\_USD\_2015 / **1000**;

depr\_sub\_2019 = Depreciation\_\_\_Amortization\_th\_U / **1000**;

depr\_sub\_2018 = Depreciation\_\_\_Amortization\_th\_1 / **1000**;

depr\_sub\_2017 = Depreciation\_\_\_Amortization\_th\_2 / **1000**;

depr\_sub\_2016 = Depreciation\_\_\_Amortization\_th\_3 / **1000**;

depr\_sub\_2015 = Depreciation\_\_\_Amortization\_th\_4 / **1000**;

other\_operating\_items\_sub\_2019 = Other\_operating\_items\_th\_USD\_201 / **1000**;

other\_operating\_items\_sub\_2018 = Other\_operating\_items\_th\_USD\_202 / **1000**;

other\_operating\_items\_sub\_2017 = Other\_operating\_items\_th\_USD\_203 / **1000**;

other\_operating\_items\_sub\_2016 = Other\_operating\_items\_th\_USD\_204 / **1000**;

other\_operating\_items\_sub\_2015 = Other\_operating\_items\_th\_USD\_205 / **1000**;

intpn\_sub\_2019 = Interest\_paid\_th\_USD\_2019 / **1000**;

intpn\_sub\_2018 = Interest\_paid\_th\_USD\_2018 / **1000**;

intpn\_sub\_2017 = Interest\_paid\_th\_USD\_2017 / **1000**;

intpn\_sub\_2016 = Interest\_paid\_th\_USD\_2016 / **1000**;

intpn\_sub\_2015 = Interest\_paid\_th\_USD\_2015 / **1000**;

xrd\_sub\_2019 = Research\_\_\_Development\_expenses / **1000**;

xrd\_sub\_2018 = Research\_\_\_Development\_expenses1 / **1000**;

xrd\_sub\_2017 = Research\_\_\_Development\_expenses2 / **1000**;

xrd\_sub\_2016 = Research\_\_\_Development\_expenses3 / **1000**;

xrd\_sub\_2015 = Research\_\_\_Development\_expenses4 / **1000**;

chech\_sub\_2019 = Cash\_flow\_th\_USD\_2019 / **1000**;

chech\_sub\_2018 = Cash\_flow\_th\_USD\_2018 / **1000**;

chech\_sub\_2017 = Cash\_flow\_th\_USD\_2017 / **1000**;

chech\_sub\_2016 = Cash\_flow\_th\_USD\_2016 / **1000**;

chech\_sub\_2015 = Cash\_flow\_th\_USD\_2015 / **1000**;

ebitda\_sub\_2019 = EBITDA\_th\_USD\_2019 / **1000**;

ebitda\_sub\_2018 = EBITDA\_th\_USD\_2018 / **1000**;

ebitda\_sub\_2017 = EBITDA\_th\_USD\_2017 / **1000**;

ebitda\_sub\_2016 = EBITDA\_th\_USD\_2016 / **1000**;

ebitda\_sub\_2015 = EBITDA\_th\_USD\_2015 / **1000**;

drop Fixed\_assets\_th\_USD\_2019;

drop Fixed\_assets\_th\_USD\_2018;

drop Fixed\_assets\_th\_USD\_2017;

drop Fixed\_assets\_th\_USD\_2016;

drop Fixed\_assets\_th\_USD\_2015;

drop Intangible\_fixed\_assets\_th\_USD\_2;

drop Intangible\_fixed\_assets\_th\_USD\_1;

drop Intangible\_fixed\_assets\_th\_USD\_3;

drop Intangible\_fixed\_assets\_th\_USD\_4;

drop Intangible\_fixed\_assets\_th\_USD\_5;

drop Tangible\_fixed\_assets\_th\_USD\_201;

drop Tangible\_fixed\_assets\_th\_USD\_202;

drop Tangible\_fixed\_assets\_th\_USD\_203;

drop Tangible\_fixed\_assets\_th\_USD\_204;

drop Tangible\_fixed\_assets\_th\_USD\_205;

drop Other\_fixed\_assets\_th\_USD\_2019;

drop Other\_fixed\_assets\_th\_USD\_2018;

drop Other\_fixed\_assets\_th\_USD\_2017;

drop Other\_fixed\_assets\_th\_USD\_2016;

drop Other\_fixed\_assets\_th\_USD\_2015;

drop Current\_assets\_th\_USD\_2019;

drop Current\_assets\_th\_USD\_2018;

drop Current\_assets\_th\_USD\_2017;

drop Current\_assets\_th\_USD\_2016;

drop Current\_assets\_th\_USD\_2015;

drop Stock\_th\_USD\_2019;

drop Stock\_th\_USD\_2018;

drop Stock\_th\_USD\_2017;

drop Stock\_th\_USD\_2016;

drop Stock\_th\_USD\_2015;

drop Debtors\_th\_USD\_2019;

drop Debtors\_th\_USD\_2018;

drop Debtors\_th\_USD\_2017;

drop Debtors\_th\_USD\_2016;

drop Debtors\_th\_USD\_2015;

drop Other\_current\_assets\_th\_USD\_2019;

drop Other\_current\_assets\_th\_USD\_2018;

drop Other\_current\_assets\_th\_USD\_2017;

drop Other\_current\_assets\_th\_USD\_2016;

drop Other\_current\_assets\_th\_USD\_2015;

drop Cash\_\_\_cash\_equivalent\_th\_USD\_20;

drop Cash\_\_\_cash\_equivalent\_th\_USD\_21;

drop Cash\_\_\_cash\_equivalent\_th\_USD\_22;

drop Cash\_\_\_cash\_equivalent\_th\_USD\_23;

drop Cash\_\_\_cash\_equivalent\_th\_USD\_24;

drop Total\_assets\_th\_USD\_2019;

drop Total\_assets\_th\_USD\_2018;

drop Total\_assets\_th\_USD\_2017;

drop Total\_assets\_th\_USD\_2016;

drop Total\_assets\_th\_USD\_2015;

drop Shareholders\_funds\_th\_USD\_2019;

drop Shareholders\_funds\_th\_USD\_2018;

drop Shareholders\_funds\_th\_USD\_2017;

drop Shareholders\_funds\_th\_USD\_2016;

drop Shareholders\_funds\_th\_USD\_2015;

drop Capital\_th\_USD\_2019;

drop Capital\_th\_USD\_2018;

drop Capital\_th\_USD\_2017;

drop Capital\_th\_USD\_2016;

drop Capital\_th\_USD\_2015;

drop Other\_shareholders\_funds\_th\_USD;

drop Other\_shareholders\_funds\_th\_USD1;

drop Other\_shareholders\_funds\_th\_USD2;

drop Other\_shareholders\_funds\_th\_USD3;

drop Other\_shareholders\_funds\_th\_USD4;

drop Non\_current\_liabilities\_th\_USD\_2;

drop Non\_current\_liabilities\_th\_USD\_1;

drop Non\_current\_liabilities\_th\_USD\_3;

drop Non\_current\_liabilities\_th\_USD\_4;

drop Non\_current\_liabilities\_th\_USD\_5;

drop Long\_term\_debt\_th\_USD\_2019;

drop Long\_term\_debt\_th\_USD\_2018;

drop Long\_term\_debt\_th\_USD\_2017;

drop Long\_term\_debt\_th\_USD\_2016;

drop Long\_term\_debt\_th\_USD\_2015;

drop Other\_non\_current\_liabilities\_th;

drop Other\_non\_current\_liabilities\_t1;

drop Other\_non\_current\_liabilities\_t2;

drop Other\_non\_current\_liabilities\_t3;

drop Other\_non\_current\_liabilities\_t4;

drop Provisions\_th\_USD\_2019;

drop Provisions\_th\_USD\_2018;

drop Provisions\_th\_USD\_2017;

drop Provisions\_th\_USD\_2016;

drop Provisions\_th\_USD\_2015;

drop Current\_liabilities\_th\_USD\_2019;

drop Current\_liabilities\_th\_USD\_2018;

drop Current\_liabilities\_th\_USD\_2017;

drop Current\_liabilities\_th\_USD\_2016;

drop Current\_liabilities\_th\_USD\_2015;

drop Loans\_th\_USD\_2019;

drop Loans\_th\_USD\_2018;

drop Loans\_th\_USD\_2017;

drop Loans\_th\_USD\_2016;

drop Loans\_th\_USD\_2015;

drop Creditors\_th\_USD\_2019;

drop Creditors\_th\_USD\_2018;

drop Creditors\_th\_USD\_2017;

drop Creditors\_th\_USD\_2016;

drop Creditors\_th\_USD\_2015;

drop Other\_current\_liabilities\_th\_USD;

drop Other\_current\_liabilities\_th\_US1;

drop Other\_current\_liabilities\_th\_US2;

drop Other\_current\_liabilities\_th\_US3;

drop Other\_current\_liabilities\_th\_US4;

drop Total\_shareh\_funds\_\_\_liab\_th\_USD;

drop Total\_shareh\_funds\_\_\_liab\_th\_US1;

drop Total\_shareh\_funds\_\_\_liab\_th\_US2;

drop Total\_shareh\_funds\_\_\_liab\_th\_US3;

drop Total\_shareh\_funds\_\_\_liab\_th\_US4;

drop Working\_capital\_th\_USD\_2019;

drop Working\_capital\_th\_USD\_2018;

drop Working\_capital\_th\_USD\_2017;

drop Working\_capital\_th\_USD\_2016;

drop Working\_capital\_th\_USD\_2015;

drop Net\_current\_assets\_th\_USD\_2019;

drop Net\_current\_assets\_th\_USD\_2018;

drop Net\_current\_assets\_th\_USD\_2017;

drop Net\_current\_assets\_th\_USD\_2016;

drop Net\_current\_assets\_th\_USD\_2015;

drop Enterprise\_value\_th\_USD\_2019;

drop Enterprise\_value\_th\_USD\_2018;

drop Enterprise\_value\_th\_USD\_2017;

drop Enterprise\_value\_th\_USD\_2016;

drop Enterprise\_value\_th\_USD\_2015;

drop Number\_of\_employees\_2019;

drop Number\_of\_employees\_2018;

drop Number\_of\_employees\_2017;

drop Number\_of\_employees\_2016;

drop Number\_of\_employees\_2015;

drop Operating\_revenue\_\_Turnover\_\_th;

drop Operating\_revenue\_\_Turnover\_\_th1;

drop Operating\_revenue\_\_Turnover\_\_th2;

drop Operating\_revenue\_\_Turnover\_\_th3;

drop Operating\_revenue\_\_Turnover\_\_th4;

drop Sales\_th\_USD\_2019;

drop Sales\_th\_USD\_2018;

drop Sales\_th\_USD\_2017;

drop Sales\_th\_USD\_2016;

drop Sales\_th\_USD\_2015;

drop Costs\_of\_goods\_sold\_th\_USD\_2019;

drop Costs\_of\_goods\_sold\_th\_USD\_2018;

drop Costs\_of\_goods\_sold\_th\_USD\_2017;

drop Costs\_of\_goods\_sold\_th\_USD\_2016;

drop Costs\_of\_goods\_sold\_th\_USD\_2015;

drop Gross\_profit\_th\_USD\_2019;

drop Gross\_profit\_th\_USD\_2018;

drop Gross\_profit\_th\_USD\_2017;

drop Gross\_profit\_th\_USD\_2016;

drop Gross\_profit\_th\_USD\_2015;

drop Other\_operating\_expenses\_th\_USD;

drop Other\_operating\_expenses\_th\_USD1;

drop Other\_operating\_expenses\_th\_USD2;

drop Other\_operating\_expenses\_th\_USD3;

drop Other\_operating\_expenses\_th\_USD4;

drop Operating\_P\_L\_\_EBIT\_th\_USD\_2019;

drop Operating\_P\_L\_\_EBIT\_th\_USD\_2018;

drop Operating\_P\_L\_\_EBIT\_th\_USD\_2017;

drop Operating\_P\_L\_\_EBIT\_th\_USD\_2016;

drop Operating\_P\_L\_\_EBIT\_th\_USD\_2015;

drop Financial\_P\_L\_th\_USD\_2019;

drop Financial\_P\_L\_th\_USD\_2018;

drop Financial\_P\_L\_th\_USD\_2017;

drop Financial\_P\_L\_th\_USD\_2016;

drop Financial\_P\_L\_th\_USD\_2015;

drop Financial\_revenue\_th\_USD\_2019;

drop Financial\_revenue\_th\_USD\_2018;

drop Financial\_revenue\_th\_USD\_2017;

drop Financial\_revenue\_th\_USD\_2016;

drop Financial\_revenue\_th\_USD\_2015;

drop Financial\_expenses\_th\_USD\_2019;

drop Financial\_expenses\_th\_USD\_2018;

drop Financial\_expenses\_th\_USD\_2017;

drop Financial\_expenses\_th\_USD\_2016;

drop Financial\_expenses\_th\_USD\_2015;

drop P\_L\_before\_tax\_th\_USD\_2019;

drop P\_L\_before\_tax\_th\_USD\_2018;

drop P\_L\_before\_tax\_th\_USD\_2017;

drop P\_L\_before\_tax\_th\_USD\_2016;

drop P\_L\_before\_tax\_th\_USD\_2015;

drop Taxation\_th\_USD\_2019;

drop Taxation\_th\_USD\_2018;

drop Taxation\_th\_USD\_2017;

drop Taxation\_th\_USD\_2016;

drop Taxation\_th\_USD\_2015;

drop P\_L\_after\_tax\_th\_USD\_2019;

drop P\_L\_after\_tax\_th\_USD\_2018;

drop P\_L\_after\_tax\_th\_USD\_2017;

drop P\_L\_after\_tax\_th\_USD\_2016;

drop P\_L\_after\_tax\_th\_USD\_2015;

drop Extr\_\_and\_other\_P\_L\_th\_USD\_2019;

drop Extr\_\_and\_other\_P\_L\_th\_USD\_2018;

drop Extr\_\_and\_other\_P\_L\_th\_USD\_2017;

drop Extr\_\_and\_other\_P\_L\_th\_USD\_2016;

drop Extr\_\_and\_other\_P\_L\_th\_USD\_2015;

drop Extr\_\_and\_other\_revenue\_th\_USD\_2;

drop Extr\_\_and\_other\_revenue\_th\_USD\_1;

drop Extr\_\_and\_other\_revenue\_th\_USD\_3;

drop Extr\_\_and\_other\_revenue\_th\_USD\_4;

drop Extr\_\_and\_other\_revenue\_th\_USD\_5;

drop Extr\_\_and\_other\_expenses\_th\_USD;

drop Extr\_\_and\_other\_expenses\_th\_USD1;

drop Extr\_\_and\_other\_expenses\_th\_USD2;

drop Extr\_\_and\_other\_expenses\_th\_USD3;

drop Extr\_\_and\_other\_expenses\_th\_USD4;

drop P\_L\_for\_period\_\_Net\_income\_th\_US;

drop P\_L\_for\_period\_\_Net\_income\_th\_U1;

drop P\_L\_for\_period\_\_Net\_income\_th\_U2;

drop P\_L\_for\_period\_\_Net\_income\_th\_U3;

drop P\_L\_for\_period\_\_Net\_income\_th\_U4;

drop Export\_revenue\_th\_USD\_2019;

drop Export\_revenue\_th\_USD\_2018;

drop Export\_revenue\_th\_USD\_2017;

drop Export\_revenue\_th\_USD\_2016;

drop Export\_revenue\_th\_USD\_2015;

drop Material\_costs\_th\_USD\_2019;

drop Material\_costs\_th\_USD\_2018;

drop Material\_costs\_th\_USD\_2017;

drop Material\_costs\_th\_USD\_2016;

drop Material\_costs\_th\_USD\_2015;

drop Costs\_of\_employees\_th\_USD\_2019;

drop Costs\_of\_employees\_th\_USD\_2018;

drop Costs\_of\_employees\_th\_USD\_2017;

drop Costs\_of\_employees\_th\_USD\_2016;

drop Costs\_of\_employees\_th\_USD\_2015;

drop Depreciation\_\_\_Amortization\_th\_U;

drop Depreciation\_\_\_Amortization\_th\_1;

drop Depreciation\_\_\_Amortization\_th\_2;

drop Depreciation\_\_\_Amortization\_th\_3;

drop Depreciation\_\_\_Amortization\_th\_4;

drop Other\_operating\_items\_th\_USD\_201;

drop Other\_operating\_items\_th\_USD\_202;

drop Other\_operating\_items\_th\_USD\_203;

drop Other\_operating\_items\_th\_USD\_204;

drop Other\_operating\_items\_th\_USD\_205;

drop Interest\_paid\_th\_USD\_2019;

drop Interest\_paid\_th\_USD\_2018;

drop Interest\_paid\_th\_USD\_2017;

drop Interest\_paid\_th\_USD\_2016;

drop Interest\_paid\_th\_USD\_2015;

drop Research\_\_\_Development\_expenses;

drop Research\_\_\_Development\_expenses1;

drop Research\_\_\_Development\_expenses2;

drop Research\_\_\_Development\_expenses3;

drop Research\_\_\_Development\_expenses4;

drop Cash\_flow\_th\_USD\_2019;

drop Cash\_flow\_th\_USD\_2018;

drop Cash\_flow\_th\_USD\_2017;

drop Cash\_flow\_th\_USD\_2016;

drop Cash\_flow\_th\_USD\_2015;

drop EBITDA\_th\_USD\_2019;

drop EBITDA\_th\_USD\_2018;

drop EBITDA\_th\_USD\_2017;

drop EBITDA\_th\_USD\_2016;

drop EBITDA\_th\_USD\_2015;

drop a;

drop auditorfirm;

**run**;

\* Save dataset to library;

**data** Caudit.orbis\_subs\_det;

set orbis\_det3;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 11. Calculate Subsidiary-level control variables

Generates the following variables (see Appendix for definitions):

N Variable Name Label in Paper

1. size\_sub Subsidiary Size

2. roa\_sub ROA

3. absroa\_sub Abs. ROA

4. revgwth\_sub Revenue Growth

5. salesvol\_sub Sales Volatility

6. leverage\_sub Leverage

7. loss\_sub Loss

8. invrec\_sub Inv-Rec

9. ifrs IFRS

10. taxhaven Tax Haven

11. EngProf English

11. rol Rule of Law

12. absaccruals\_sub Abs. Total Accruals

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Orbis provides data collapsed at the subsidiary level by identify BVDID (i.e., all variables across all time periods are on one line for each subsidiary).

In this step we transpose the Orbis Dataset by Global Ultimate Owner (GUO) cik, country, subsidiary bvd\_id (a separate transform procedure must be performed for each variable);

**data** s1;

set Caudit.orbis\_subs\_det;

if missing(cik) then delete;

rename Country\_ISO\_code = country;

rename Last\_avail\_year = guo\_year;

**run**;

**proc** **sort** data=s1 nodupkey; by cik bvdid country sic3 orbis\_guo\_year stat\_auditor ; **run**;

**proc** **transpose** data=s1 out=Tr1 (rename=(\_NAME\_ = transpose COL1 = ca\_raw));

var ca\_sub\_2019 ca\_sub\_2018 ca\_sub\_2017 ca\_sub\_2016 ca\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr1a;

set tr1;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ca\_sub **12.**;

ca\_sub = ca\_raw;

drop ca\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr2 (rename=(\_NAME\_ = transpose COL1 = cash\_raw));

var cash\_sub\_2019 cash\_sub\_2018 cash\_sub\_2017 cash\_sub\_2016 cash\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr2a;

set tr2;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format cash\_sub **8.**;

cash\_sub = cash\_raw;

drop cash\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr3 (rename=(\_NAME\_ = transpose COL1 = at\_raw));

var at\_sub\_2019 at\_sub\_2018 at\_sub\_2017 at\_sub\_2016 at\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr3a;

set tr3;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format at\_sub **8.**;

at\_sub = at\_raw;

drop at\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr4 (rename=(\_NAME\_ = transpose COL1 = ppe\_raw));

var ppe\_sub\_2019 ppe\_sub\_2018 ppe\_sub\_2017 ppe\_sub\_2016 ppe\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr4a;

set tr4;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ppe\_sub **8.**;

ppe\_sub = ppe\_raw;

drop ppe\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr5 (rename=(\_NAME\_ = transpose COL1 = cl\_raw));

var cl\_sub\_2019 cl\_sub\_2018 cl\_sub\_2017 cl\_sub\_2016 cl\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr5a;

set tr5;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format cl\_sub **8.**;

cl\_sub = cl\_raw;

drop cl\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr6 (rename=(\_NAME\_ = transpose COL1 = ib\_raw));

var ib\_sub\_2019 ib\_sub\_2018 ib\_sub\_2017 ib\_sub\_2016 ib\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr6a;

set tr6;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ib\_sub **8.**;

ib\_sub = ib\_raw;

drop ib\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr7 (rename=(\_NAME\_ = transpose COL1 = sale\_raw));

var sale\_sub\_2019 sale\_sub\_2018 sale\_sub\_2017 sale\_sub\_2016 sale\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr7a;

set tr7;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format sale\_sub **8.**;

sale\_sub = sale\_raw;

drop sale\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr8 (rename=(\_NAME\_ = transpose COL1 = ebit\_raw));

var ebit\_sub\_2019 ebit\_sub\_2018 ebit\_sub\_2017 ebit\_sub\_2016 ebit\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr8a;

set tr8;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ebit\_sub **8.**;

ebit\_sub = ebit\_raw;

drop ebit\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr9 (rename=(\_NAME\_ = transpose COL1 = ni\_raw));

var ni\_sub\_2019 ni\_sub\_2018 ni\_sub\_2017 ni\_sub\_2016 ni\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr9a;

set tr9;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ni\_sub **8.**;

ni\_sub = ni\_raw;

drop ni\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr10 (rename=(\_NAME\_ = transpose COL1 = numemp\_raw));

var numemp\_sub\_2019 numemp\_sub\_2018 numemp\_sub\_2017 numemp\_sub\_2016 numemp\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr10a;

set tr10;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format numemp\_sub **8.**;

numemp\_sub = numemp\_raw;

drop numemp\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr11 (rename=(\_NAME\_ = transpose COL1 = chech\_raw));

var chech\_sub\_2019 chech\_sub\_2018 chech\_sub\_2017 chech\_sub\_2016 chech\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr11a;

set tr11;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format chech\_sub **8.**;

chech\_sub = chech\_raw;

drop chech\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr12 (rename=(\_NAME\_ = transpose COL1 = ebt\_raw));

var ebt\_sub\_2019 ebt\_sub\_2018 ebt\_sub\_2017 ebt\_sub\_2016 ebt\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr12a;

set tr12;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ebt\_sub **8.**;

ebt\_sub = ebt\_raw;

drop ebt\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr13 (rename=(\_NAME\_ = transpose COL1 = equity\_raw));

var equity\_sub\_2019 equity\_sub\_2018 equity\_sub\_2017 equity\_sub\_2016 equity\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr13a;

set tr13;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format equity\_sub **8.**;

equity\_sub = equity\_raw;

drop equity\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr14 (rename=(\_NAME\_ = transpose COL1 = depr\_raw));

var depr\_sub\_2019 depr\_sub\_2018 depr\_sub\_2017 depr\_sub\_2016 depr\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr14a;

set tr14;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format depr\_sub **8.**;

depr\_sub = depr\_raw;

drop depr\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr15 (rename=(\_NAME\_ = transpose COL1 = ap\_raw));

var ap\_sub\_2019 ap\_sub\_2018 ap\_sub\_2017 ap\_sub\_2016 ap\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr15a;

set Tr15;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ap\_sub **12.**;

ap\_sub = ap\_raw;

drop ap\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr16 (rename=(\_NAME\_ = transpose COL1 = ar\_raw));

var ar\_sub\_2019 ar\_sub\_2018 ar\_sub\_2017 ar\_sub\_2016 ar\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr16a;

set Tr16;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ar\_sub **12.**;

ar\_sub = ar\_raw;

drop ar\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr17 (rename=(\_NAME\_ = transpose COL1 = cogs\_raw));

var cogs\_sub\_2019 cogs\_sub\_2018 cogs\_sub\_2017 cogs\_sub\_2016 cogs\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr17a;

set Tr17;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format cogs\_sub **12.**;

cogs\_sub = cogs\_raw;

drop cogs\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr18 (rename=(\_NAME\_ = transpose COL1 = dltt\_raw));

var dltt\_sub\_2019 dltt\_sub\_2018 dltt\_sub\_2017 dltt\_sub\_2016 dltt\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr18a;

set Tr18;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format dltt\_sub **12.**;

dltt\_sub = dltt\_raw;

drop dltt\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr19 (rename=(\_NAME\_ = transpose COL1 = ebitda\_raw));

var ebitda\_sub\_2019 ebitda\_sub\_2018 ebitda\_sub\_2017 ebitda\_sub\_2016 ebitda\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr19a;

set Tr19;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format ebitda\_sub **12.**;

ebitda\_sub = ebitda\_raw;

drop ebitda\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr20 (rename=(\_NAME\_ = transpose COL1 = equity\_other\_raw));

var equity\_other\_sub\_2019 equity\_other\_sub\_2018 equity\_other\_sub\_2017 equity\_other\_sub\_2016 equity\_other\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr20a;

set Tr20;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format equity\_other\_sub **12.**;

equity\_other\_sub = equity\_other\_raw;

drop equity\_other\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr21 (rename=(\_NAME\_ = transpose COL1 = intang\_\_raw));

var intang\_\_sub\_2019 intang\_\_sub\_2018 intang\_\_sub\_2017 intang\_\_sub\_2016 intang\_\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr21a;

set Tr21;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format intang\_\_sub **12.**;

intang\_\_sub = intang\_\_raw;

drop intang\_\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr22 (rename=(\_NAME\_ = transpose COL1 = inv\_raw));

var inv\_sub\_2019 inv\_sub\_2018 inv\_sub\_2017 inv\_sub\_2016 inv\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr22a;

set Tr22;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format inv\_sub **12.**;

inv\_sub = inv\_raw;

drop inv\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr23 (rename=(\_NAME\_ = transpose COL1 = net\_ca\_raw));

var net\_ca\_sub\_2019 net\_ca\_sub\_2018 net\_ca\_sub\_2017 net\_ca\_sub\_2016 net\_ca\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr23a;

set Tr23;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format net\_ca\_sub **12.**;

net\_ca\_sub = net\_ca\_raw;

drop net\_ca\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr24 (rename=(\_NAME\_ = transpose COL1 = noncurr\_lt\_raw));

var noncurr\_lt\_sub\_2019 noncurr\_lt\_sub\_2018 noncurr\_lt\_sub\_2017 noncurr\_lt\_sub\_2016 noncurr\_lt\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr24a;

set Tr24;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format noncurr\_lt\_sub **12.**;

noncurr\_lt\_sub = noncurr\_lt\_raw;

drop noncurr\_lt\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr25 (rename=(\_NAME\_ = transpose COL1 = oprev\_raw));

var oprev\_sub\_2019 oprev\_sub\_2018 oprev\_sub\_2017 oprev\_sub\_2016 oprev\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr25a;

set Tr25;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format oprev\_sub **12.**;

oprev\_sub = oprev\_raw;

drop oprev\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr26 (rename=(\_NAME\_ = transpose COL1 = std\_raw));

var std\_sub\_2019 std\_sub\_2018 std\_sub\_2017 std\_sub\_2016 std\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr26a;

set Tr26;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format std\_sub **12.**;

std\_sub = std\_raw;

drop std\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr27 (rename=(\_NAME\_ = transpose COL1 = txt\_raw));

var txt\_sub\_2019 txt\_sub\_2018 txt\_sub\_2017 txt\_sub\_2016 txt\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr27a;

set Tr27;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format txt\_sub **12.**;

txt\_sub = txt\_raw;

drop txt\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr28 (rename=(\_NAME\_ = transpose COL1 = wcap\_raw));

var wcap\_sub\_2019 wcap\_sub\_2018 wcap\_sub\_2017 wcap\_sub\_2016 wcap\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr28a;

set Tr28;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format wcap\_sub **12.**;

wcap\_sub = wcap\_raw;

drop wcap\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr29 (rename=(\_NAME\_ = transpose COL1 = xrd\_raw));

var xrd\_sub\_2019 xrd\_sub\_2018 xrd\_sub\_2017 xrd\_sub\_2016 xrd\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr29a;

set Tr29;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format xrd\_sub **12.**;

xrd\_sub = xrd\_raw;

drop xrd\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr30 (rename=(\_NAME\_ = transpose COL1 = other\_cl\_raw));

var other\_cl\_sub\_2019 other\_cl\_sub\_2018 other\_cl\_sub\_2017 other\_cl\_sub\_2016 other\_cl\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr30a;

set Tr30;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format other\_cl\_sub **12.**;

other\_cl\_sub = other\_cl\_raw;

drop other\_cl\_raw;

**run**;

**proc** **transpose** data=s1 out=Tr31 (rename=(\_NAME\_ = transpose COL1 = oca\_raw));

var oca\_sub\_2019 oca\_sub\_2018 oca\_sub\_2017 oca\_sub\_2016 oca\_sub\_2015;

by cik bvdid country sic3 orbis\_guo\_year stat\_auditor;

**run**;

**data** Tr31a;

set Tr31;

format year **8.**;

year = substr(transpose,length(transpose)-**3**); drop transpose;

format oca\_sub **12.**;

oca\_sub = oca\_raw;

drop oca\_raw;

**run**;

**data** s2;

merge tr1a tr2a tr3a tr4a tr5a tr6a tr7a tr8a tr9a tr10a

tr11a tr12a tr13a tr14a tr15a tr16a tr17a tr18a tr19a

tr20a tr21a tr22a tr23a tr24a tr25a tr26a tr27a tr28a

tr29a tr30a tr31a;

sub = **1**;

if missing(country) then delete;

**run**;

**proc** **sort** data=s2; by cik year country; **run**;

\* Save transposed dataset to library;

**data** Caudit.orbis\_subs\_tr;

set s2;

**run**;

\* check data;

**proc** **means** data=s2 n nmiss mean median min max; **run**;

**proc** **sort** data=Caudit.orbis\_subs\_tr out=o0a nodupkey; by cik bvdid year; **run**;

\* Generate Subsidiary Variables from Orbis;

\* Note - "change" is the annual raw change - "delta" is the weighted/final variable

for any changes;

**data** o1;

set o0a;

if missing(cik) then delete;

\* set lags of variables;

lagcik = lag(cik);

lagbvdid = lag(bvdid);

lagyear = lag(year);

lagat\_sub = lag(at\_sub);

lagoprev\_sub = lag(oprev\_sub);

lagca\_sub = lag(ca\_sub);

lagcl\_sub = lag(cl\_sub);

lagcash\_sub = lag(cash\_sub);

lagstd\_sub = lag(std\_sub);

if lagcik ne cik or lagbvdid ne bvdid or lagyear ne year - **1** then do;

lagat\_sub = **.**;

lagoprev\_sub = **.**;

lagca\_sub = **.**;

lagcl\_sub = **.**;

lagcash\_sub = **.**;

lagstd\_sub = **.**;

end;

lag2cik = lag2(cik);

lag2bvdid = lag2(bvdid);

lag2year = lag2(year);

lag2at\_sub = lag2(at\_sub);

lag2oprev\_sub = lag2(oprev\_sub);

if lag2cik ne cik or lag2bvdid ne bvdid or lag2year ne year - **2** then do;

lag2at\_sub = **.**;

lag2oprev\_sub = **.**;

end;

\* Calculate Control Variables;

\* Size - log of subsidiary assets;

size\_sub = log(at\_sub);

\* ROA;

avgat\_sub = (at\_sub + lagat\_sub) / **2**;

roa\_sub = ib\_sub / avgat\_sub;

\* Revenue Growth;

revgwth\_sub = (oprev\_sub - lagoprev\_sub) / avgat\_sub;

\* Sales Volatility;

s1 = oprev\_sub / at\_sub;

s2 = lagoprev\_sub / lagat\_sub;

s3 = lag2oprev\_sub / lag2at\_sub;

if s1 ne **.** and s2 ne **.** then salesvol\_sub = std(s1,s2,s3);

\* Leverage;

leverage\_sub = (dltt\_sub + std\_sub) / at\_sub;

\* Loss;

if ni\_sub ne **.** and ni\_sub < **0** then loss\_sub = **1**; else if ni\_sub ne **.** then loss\_sub = **0**;

\* Inv-Rec - inventory and receivables intensity;

invrec\_sub = (inv\_sub + ar\_sub) / at\_sub;

if invrec\_sub = **.** then invrec\_sub = **0**;

\* Calculate variables for abnormal accruals calculation (for use in STEP 13);

\* Calculate accruals, following Beuselinck et al. (2019);

change\_cash = cash\_sub - lagcash\_sub;

change\_ca = ca\_sub - lagca\_sub;

change\_cl = cl\_sub - lagcl\_sub;

change\_std = std\_sub = lagstd\_sub;

accruals\_raw = ((change\_ca - change\_cash) - (change\_cl - change\_std) - depr\_sub);

accruals\_sub = accruals\_raw / avgat\_sub;

absaccruals\_sub = abs(accruals\_sub);

ppe\_at\_sub = ppe\_sub / avgat\_sub;

invat\_sub = **1** / lagat\_sub;

\* Mark financial and utilities companies;

if SIC3 >= **600** and SIC3 <= **699** then finance = **1**; else finance = **0**;

if SIC3 >= **490** and SIC3 <= **494** then utility = **1**; else utility = **0**;

\* Generate 2-Digit industry variables;

sic2 = int(sic3 / **10**);

**run**;

\* check data;

**proc** **means** data=o1 n nmiss mean min median max; **run**;

\* Merge in Country-level variables

Country-level data, such as English proficiency, were hand-collected from the sources documented in Appendix A;

**data** country;

set Caudit.countrycode (rename = (id = countryid isocountrycode = country));

keep countryname region countryid wgicode english ef\_full ifrs pcaob\_insp country taxhaven;

**run**;

**proc** **sort** data=country; by country; **run**;

**proc** **sort** data=o1; by country; **run**;

\*Merge and convert English proficiency to numeric;

**data** o2;

merge o1 (in=xx) country (in = yy);

by country;

if xx;

EngProf = ef\_full\***1**;

drop ef\_full;

**run**;

\* Merge Rule of Law data from World Governance Indicators (World Bank 2019);

\* WGI data only goes through 2017 (as of data collection) - we extend 2017 ROL to 2018 and 2019 observations;

**data** rol;

set Caudit.wgi;

**run**;

**data** rol18;

set Caudit.wgi;

where year = **2017**;

year = year+**1**;

**run**;

**data** rol19;

set Caudit.wgi;

where year = **2017**;

year = year+**2**;

**run**;

\* append ROL18 and 19 data to ROL;

**proc** **append** base=rol data=rol18; **run**;

**proc** **append** base=rol data=rol19; **run**;

\* add ROL to dataset;

**proc** **sql**;

create table o3 as

select a.\*, b.rle as rol

from o2 as a left join rol as b

on a.wgicode = b.code and a.year = b.year;

**quit**;

\* Winsorize Continuous Variables;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Winsorize macro

Can winsorize or trim at specified percentiles;

dsetout = leave blank to overwrite dsetin

byvar = none for no byvar (trims/winsorizes pooled sample)

type = delete/winsor

ex: %winsor(dsetin=mydata, dsetout=mydata2, byvar=year, vars=assets earnings, pctl=0 98);

winsorizes by year at 98%, puts resulting dataset into mydata2

%winsor(dsetin=mydata, vars=assets earnings, type=delete);

trims pooled sample at 1% and 99%, puts resulting dataset back into mydata

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\* Winsorize as last step;

**%macro** winsor(dsetin=o3, dsetout =, byvar=none, vars=

size\_sub roa\_sub revgwth\_sub salesvol\_sub leverage\_sub invrec\_sub absaccruals\_sub

rol engprof,

type=winsor, pctl=**1** **99**);

\*nothing gets changed below\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

%if &dsetout = %then %let dsetout = &dsetin;

%let varL=;

%let varH=;

%let xn=1;

%do %until ( %scan(&vars,&xn)= );

%let token = %scan(&vars,&xn);

%let varL = &varL &token.L;

%let varH = &varH &token.H;

%let xn=%EVAL(&xn + 1);

%end;

%let xn=%eval(&xn-1);

data xtemp;

set &dsetin;

run;

%if &byvar = none %then %do;

data xtemp;

set xtemp;

xbyvar = **1**;

run;

%let byvar = xbyvar;

%end;

proc sort data = xtemp; by &byvar; run;

proc univariate data = xtemp noprint;

by &byvar;

var &vars;

output out = xtemp\_pctl PCTLPTS = &pctl PCTLPRE = &vars PCTLNAME = L H;

run;

data &dsetout;

merge xtemp xtemp\_pctl;

by &byvar;

array trimvars{&xn} &vars;

array trimvarl{&xn} &varL;

array trimvarh{&xn} &varH;

do xi = **1** to dim(trimvars);

%if &type = winsor %then %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then trimvars{xi} = trimvarl{xi};

if (trimvars{xi} > trimvarh{xi}) then trimvars{xi} = trimvarh{xi};

end;

%end;

%else %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then delete;

if (trimvars{xi} > trimvarh{xi}) then delete;

end;

%end;

end;

drop &varL &varH xbyvar xi;

run;

**%mend** winsor;

%***winsor***

**run**; \*executes the macro;

\* Save dataset to library;

**data** Caudit.orbis\_detail\_final;

set o3;

\* Calculate Abs. ROA;

absroa\_sub = abs(roa\_sub);

**run**;

**proc** **means** data=Caudit.orbis\_detail\_final n nmiss mean min median max; **run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 12. Merge subsidiary data with Form AP component auditor data and calculate additional variables

Generates the following variables (see Appendix for definitions):

N Variable Name Label in Paper

1. statutory\_auditor Statutory Auditor

2. MNC\_size MNC Size

3. MNC\_lnforseg MNC For. Segments

4. big6 Big 6

5. lagm302\_sub Prior MW

6. match Component Auditor

7. salesperc Subsidiary Sales-%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This step merges information from the following datasets together:

1. Orbis subsidiary details

2. Form AP - MNC level

3. Form AP - Component Auditor Level

4. Compustat - MNC-level characteristics

5. Hand-collected subsidiary MW data

It also matches Orbis subsidiary observations to specific individually identified component auditors.;

\* load Orbis sample;

**data** o0;

set Caudit.orbis\_detail\_final;

rename regioncode = regionid;

cik2 = cik\***1**;

if year > **2016**;

**run**;

\* Check unique Subs;

**proc** **sort** data=o0 out=check1 nodupkey; by bvdid; **run**;

\* Add most recent data collection date from Orbis (separately downloaded);

**proc** **sort** data=Caudit.orbis\_lastavail out=avail; by bvdid; **run**;

**data** avail;

set avail;

format orbis\_guo\_year **8.**;

orbis\_guo\_year = guo\_year \* **1**;

**run**;

**proc** **sql**;

create table o1 as

select a.\*, b.orbis\_guo\_year

from o0 as a left join avail as b

on a.bvdid = b.bvdid;

**quit**;

**proc** **means** data=o1 n nmiss mean min max; **run**;

\* Merge Current Statutory Auditors with dataset - downloaded separately from Orbis;

**proc** **sort** data=Caudit.orbis\_auditors out=stat; by BvD\_ID\_number; **run**;

**data** stat;

set stat;

if ADV\_\_Current\_or\_previous = "Current";

rename BvD\_ID\_number = bvdid;

rename ADV\_\_Full\_name = stat\_auditor;

apptyear = year(ADV\_\_Appointment\_date);

**run**;

**proc** **sort** data=stat; by bvdid descending apptyear; **run**;

**proc** **sort** data=stat nodupkey; by bvdid; **run**;

**proc** **sql**;

create table o1a as

select a.\*, b.stat\_auditor

from o1 as a left join stat as b

on a.bvdid = b.bvdid;

**quit**;

\*check GUO-country obs;

**proc** **sort** data=o1a out=check1 nodupkey; by cik country; **run**;

\* Delete observations missing Orbis GUO collection dates

or with dates < 2017 (before sample period)

define Statutory Auditor;

**data** o2;

set o1a;

if **2017** > orbis\_guo\_year and orbis\_guo\_year ne **.** and at\_sub = **.** then delete;

if orbis\_guo\_year = **.** and at\_sub = **.** then delete;

if missing(stat\_auditor) then statutory\_auditor = **0**; else statutory\_auditor = **1**;

**run**;

\*check unique bvdid obs;

**proc** **sort** data=o2 out=check1 nodupkey; by bvdid; **run**;

\* Merge with MNC-level Form AP data and delete missing obs;

**proc** **sql**;

create table ap1 as

select a.\*, b.Firm\_ID, b.Firm\_country from

o2 as a left join Caudit.F1 as b

on a.cik2 = b.fkey and a.year = b.year;

**quit**;

\* Drop observations that don't merge with Form AP;

**data** ap2;

set ap1;

if firm\_country ne "";

**run**;

\* Check unique BVDIDs that match with Form AP;

**proc** **sort** data=ap2 out=check1 nodupkey; by bvdid; **run**;

\* Drop observations where Principal Auditor is outside the U.S.;

**data** ap3;

set ap2;

if firm\_country = "United States";

**run**;

\* Check unique BVDIDs and GUOs that remain;

**proc** **sort** data=ap3 out=check1 nodupkey; by bvdid; **run**;

**proc** **sort** data=ap3 out=check2 nodupkey; by cik year; **run**;

**proc** **sort** data=ap3 out=check3 nodupkey; by cik; **run**;

\* Match Component Auditors to Subsidiary Observations by CIK, Country, Year;

\* Match with component auditors by available GUO data;

\* prepare foreign component auditor country data;

**proc** **sort** data=Caudit.cmpcountries out=cmp1 nodupkey; by fkey cmp\_countryid year; **run**;

\* Available component auditor countries;

**proc** **sql**;

create table cmp2 as

select a.\*, b.cik

from cmp1 a left join ap3 b

on a.fkey = b.cik2 and a.year = b.year;

**quit**;

\* Delete duplicates after merge;

**proc** **sort** data=cmp2 nodupkey; by fkey cmp\_countryid year; **run**;

\* Total number of Foreign Component Auditor Countries to be matched;

**data** cmp3;

set cmp2;

if missing(cik) then delete;

if missing(fkey) then delete;

drop cik;

**run**;

\* Check number of unique engagement years;

**proc** **sort** data=cmp3 out=check nodupkey; by fkey year; **run**;

\* Check number of unique MNCs;

**proc** **sort** data=cmp3 out=check nodupkey; by fkey; **run**;

\* Merge in country level matches;

**proc** **sql**;

create table m1 as

select a.\*, b.cmp\_countryid, b.cmp\_hrs, b.year as cmp\_year, b.fkey as cmp\_fkey, b.cmp\_firmid

from ap3 a full join cmp3 b

on a.cik2 = b.fkey and a.countryid = b.cmp\_countryid and a.year = b.year;

**quit**;

\* Drop all non-matched observations from the main sample, and create component sample of unmatched component auditors;

**data** m2;

set m1;

if cik2 = **.** then delete;

**run**;

\* Generate dataset of unmatched Foreign Component Auditor Countries;

**data** m2\_cmp;

set m1;

if cik2 = **.**;

keep cmp\_fkey cmp\_year cmp\_countryid cmp\_hrs cmp\_firmid;

**run**;

\*Check unique issuer-years of matches;

**data** m3\_cmp;

set m2;

if cmp\_countryid ne **.**;

**run**;

\*Add firm-level Compustat data to Orbis Country Data;

**proc** **sort** data=Caudit.C1 (keep=datadate gvkey year cik at sale MNC\_lnforseg MNC\_size) out=cik nodupkey;by cik datadate;**run**;

**proc** **sql**;

create table M3 as

select a.\*, b.gvkey, b.sale, b.MNC\_lnforseg, b.MNC\_size, b.datadate

from M2 a left join cik b

on a.cik=b.cik and a.year=b.year;

**quit**;

**data** M4;

set M3;

if cik2 = **.** then delete;

rename oprev\_sub = sales;

rename at\_sub = assets;

if datadate > **'31DEC2019'd** then delete;

if MNC\_lnforseg = **.** then MNC\_lnforseg = **0**;

**run**;

\*\*\* Import Susbsidiary MW Data from Excel Import (Hand-Collected)\*\*\*\*;

**data** mw;

set Caudit.sub\_country\_mw;

year = year(datadate);

**run**;

**proc** **sql**;

create table M5 as

select a.\*, b.lagm302\_sub

from M4 a left join mw b

on a.cik2 = b.cik2 and a.year = b.year and a.countryid = b.countryid;

**quit**;

**data** M5;

set M5;

if lagm302\_sub = **.** then lagm302\_sub = **0**;

**run**;

\* delete duplicates;

**proc** **sort** data=M5; by cik2 year bvdid descending lagm302\_sub; **run**;

**proc** **sort** data=M5 nodupkey; by cik2 year bvdid; **run**;

\* Drop observations missing certain data;

\* Drop observations missing assets data ;

**data** M5a;

set M5;

if assets = **.** then delete;

**run**;

\* Unique subsidiaries contributing - 35,247;

**proc** **sort** data=M5a out=check1 nodupkey; by bvdid; **run**;

\* Drop observations with less than $1m in assets;

**data** M5b;

set M5a;

if assets < **1** then delete;

**run**;

\* Check number of unique subsidiaries contributing;

**proc** **sort** data=M5b out=check1 nodupkey; by bvdid; **run**;

\* Drop assets missing sales data;

**data** M5c;

set M5b;

if sales = **.** then delete;

**run**;

\* Check number of unique subsidiaries contributing;

**proc** **sort** data=M5c out=check1 nodupkey; by bvdid; **run**;

\* Drop observations missing MNC asset data or sales data;

**data** M5d;

set M5c;

if MNC\_size = **.** or MNC\_size = **0** or sale = **.** or sale = **0** then delete;

**run**;

\* Check number of unique subsidiaries contributing;

**proc** **sort** data=M5d out=check1 nodupkey; by bvdid; **run**;

\* Calculate Component Auditor variable of interest and control variables;

**data** M6;

set M5d;

\* Define Component Auditor variable (match);

match = **0**;

if cmp\_countryid ne **.** then match = **1**;

\* Calculate subsidiary sales as a percentage of MNC parent sales, cap at 0 and 1;

salesperc = sales / sale;

if salesperc ne **.** and salesperc < **0** then salesperc = **0**;

if salesperc ne **.** and salesperc > **1** then salesperc = **1**;

**run**;

**proc** **means** data=M6 n nmiss mean min median max; **run**;

\* Add Big 6 variable from Compustat;

**data** auditor;

set Caudit.Comp\_data1;

if au < **8** then big4 = **1**;else big4 = **0**;

mid2 = **0**;

if au = **17** then mid2 = **1**;\*Grant Thorton;

if au = **11** then mid2 = **1**;\*BDO;

big6 = **0**;

if big4 = **1** then big6 = **1**;

if mid2 = **1** then big6 = **1**;

if au = **.** then do;

big4 = **.**;

big6 = **.**;

mid2 = **.**;

end;

keep gvkey year datadate au big4 mid2 big6;

**run**;

\* Merge Big6 with subsidiary data;

**proc** **sql**;

create table M7 as

select a.\*,b.au, b.big6

from M6 a left join auditor b

on a.gvkey=b.gvkey and a.year=b.year;

**quit**;

\* Check data;

**proc** **means** data=M7 n nmiss mean median min max; **run**;

\*Save dataset to library;

**data** Caudit.orbis\_variables;

set M7;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 13 Calculate Jones (1991) abnormal accruals for subsidiaries

Generates the following variables (see Appendix for definitions):

N Variable Name Label in Paper

1. absda\_fe Abs. Abnormal Accruals

2. da\_fe used in regressions of income-increasing and income-decreasing abnormal accruals

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*Set lags and calculate main variables, clean data, winsorize;

**proc** **sort** data=Caudit.orbis\_variables out=orb nodupkey; by bvdid year; **run**;

**data** C;

set orb;

rename accruals\_sub = accruals;

rename avgat\_sub = avgat;

rename ppe\_at\_sub = ppe;

rename roa\_sub = roa;

rename invat\_sub = invat;

rename revgwth\_sub = revgwth;

**run**;

**data** C0;

set C;

keep cik bvdid year sic2 absaccruals\_sub accruals avgat ppe revgwth roa invat countryid;

**run**;

**proc** **means** data=C0 n nmiss mean min max;**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Winsorize macro

Can winsorize or trim at specified percentiles;

dsetout = leave blank to overwrite dsetin

byvar = none for no byvar (trims/winsorizes pooled sample)

type = delete/winsor

ex: %winsor(dsetin=mydata, dsetout=mydata2, byvar=year, vars=assets earnings, pctl=0 98);

winsorizes by year at 98%, puts resulting dataset into mydata2

%winsor(dsetin=mydata, vars=assets earnings, type=delete);

trims pooled sample at 1% and 99%, puts resulting dataset back into mydata

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**%macro** winsor(dsetin=C0, dsetout = , byvar=none, vars=

accruals revgwth invat ppe roa,

type=winsor, pctl=**1** **99**);

\*nothing gets changed below\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

%if &dsetout = %then %let dsetout = &dsetin;

%let varL=;

%let varH=;

%let xn=1;

%do %until ( %scan(&vars,&xn)= );

%let token = %scan(&vars,&xn);

%let varL = &varL &token.L;

%let varH = &varH &token.H;

%let xn=%EVAL(&xn + 1);

%end;

%let xn=%eval(&xn-1);

data xtemp;

set &dsetin;

run;

%if &byvar = none %then %do;

data xtemp;

set xtemp;

xbyvar = **1**;

run;

%let byvar = xbyvar;

%end;

proc sort data = xtemp; by &byvar; run;

proc univariate data = xtemp noprint;

by &byvar;

var &vars;

output out = xtemp\_pctl PCTLPTS = &pctl PCTLPRE = &vars PCTLNAME = L H;

run;

data &dsetout;

merge xtemp xtemp\_pctl;

by &byvar;

array trimvars{&xn} &vars;

array trimvarl{&xn} &varL;

array trimvarh{&xn} &varH;

do xi = **1** to dim(trimvars);

%if &type = winsor %then %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then trimvars{xi} = trimvarl{xi};

if (trimvars{xi} > trimvarh{xi}) then trimvars{xi} = trimvarh{xi};

end;

%end;

%else %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then delete;

if (trimvars{xi} > trimvarh{xi}) then delete;

end;

%end;

end;

drop &varL &varH xbyvar xi;

run;

**%mend** winsor;

%***winsor***

**run**; \*executes the macro;

**proc** **sort** data=C0;by sic2 year;**run**;

\* Regress accruals on factors from Jones (1991) and Kothari et al. (2005), including Country fixed effects, by SIC2 and Year;

**proc** **surveyreg** data=C0 ;

class countryid;

by sic2 year;

model accruals = invat revgwth ppe roa countryid ;

output out=C1 r=da\_fe;

**run**;

**data** C2;

set C1;

absda\_fe = abs(da\_fe);

**run**;

\* Check observations by industry and year - keep only variables with at least 10 total observations per industry and country;

**proc** **sort** data=c2; by sic2 year;

**proc** **means** data=C2 n noprint;

by sic2 year;

var absda\_fe;

output out=n\_sic2\_year n=n\_sic2\_year;

**run**;

**data** C3;

merge C2 n\_sic2\_year;

by sic2 year;

drop \_type\_ \_freq\_;

**run**;

**proc** **sort** data=C3; by countryid year;

**proc** **means** data=C3 n noprint;

by countryid year;

var absda\_fe;

output out=n\_country\_year n=n\_country\_year;

**run**;

**data** C4;

merge C3 n\_country\_year;

by countryid year;

drop \_type\_ \_freq\_;

**run**;

**proc** **means** data=C4 n nmiss mean median min max;**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Winsorize macro

Can winsorize or trim at specified percentiles;

dsetout = leave blank to overwrite dsetin

byvar = none for no byvar (trims/winsorizes pooled sample)

type = delete/winsor

ex: %winsor(dsetin=mydata, dsetout=mydata2, byvar=year, vars=assets earnings, pctl=0 98);

winsorizes by year at 98%, puts resulting dataset into mydata2

%winsor(dsetin=mydata, vars=assets earnings, type=delete);

trims pooled sample at 1% and 99%, puts resulting dataset back into mydata

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**%macro** winsor(dsetin=C4, dsetout = , byvar=none, vars=

absaccruals\_sub da\_fe absda\_fe,

type=winsor, pctl=**1** **99**);

\*nothing gets changed below\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

%if &dsetout = %then %let dsetout = &dsetin;

%let varL=;

%let varH=;

%let xn=1;

%do %until ( %scan(&vars,&xn)= );

%let token = %scan(&vars,&xn);

%let varL = &varL &token.L;

%let varH = &varH &token.H;

%let xn=%EVAL(&xn + 1);

%end;

%let xn=%eval(&xn-1);

data xtemp;

set &dsetin;

run;

%if &byvar = none %then %do;

data xtemp;

set xtemp;

xbyvar = **1**;

run;

%let byvar = xbyvar;

%end;

proc sort data = xtemp; by &byvar; run;

proc univariate data = xtemp noprint;

by &byvar;

var &vars;

output out = xtemp\_pctl PCTLPTS = &pctl PCTLPRE = &vars PCTLNAME = L H;

run;

data &dsetout;

merge xtemp xtemp\_pctl;

by &byvar;

array trimvars{&xn} &vars;

array trimvarl{&xn} &varL;

array trimvarh{&xn} &varH;

do xi = **1** to dim(trimvars);

%if &type = winsor %then %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then trimvars{xi} = trimvarl{xi};

if (trimvars{xi} > trimvarh{xi}) then trimvars{xi} = trimvarh{xi};

end;

%end;

%else %do;

if not missing(trimvars{xi}) then do;

if (trimvars{xi} < trimvarl{xi}) then delete;

if (trimvars{xi} > trimvarh{xi}) then delete;

end;

%end;

end;

drop &varL &varH xbyvar xi;

run;

**%mend** winsor;

%***winsor***

**run**; \*executes the macro;

**data** Caudit.subs\_absda;

set C4;

keep cik bvdid year absaccruals\_sub da\_fe absda\_fe ;

**run**;

\* Combine abnormal accruals data with main dataset;

**data** orbis;

set Caudit.orbis\_variables;

drop absaccruals\_sub;

**run**;

**proc** **sql**;

create table orbis2 as

select a.\*, b.absaccruals\_sub, b.absda\_fe, b.da\_fe, b.n\_sic2\_year, b.n\_country\_year

from orbis as a left join Caudit.subs\_absda as b

on

a.bvdid = b.bvdid and

a.year = b.year;

**quit**;

\* Identify Largest subsidiary observation by country - 17,716;

**proc** **sort** data=orbis2 nodupkey; by cik2 year countryid bvdid; **run**;

**proc** **sort** data=orbis2 out=orbis3; by cik2 year countryid descending assets; **run**;

**data** orbis4;

set orbis3;

by cik2 year countryid descending assets;

if first.countryid then largest = **1**; else largest = **0**;

**run**;

**data** orbis5;

set orbis4;

if largest = **1**;

**run**;

\* Save final dataset to library;

**data** Caudit.orbis\_final\_allsubs;

set orbis5;

\* Generate Size polynomials;

size2 = size\_sub\*\***2**;

size3 = size\_sub\*\***3**;

\* Replace Accruals variables with missing values for Finance and Utilities industries

This effectively drops these observations for the Accruals analyses;

if finance = **1** or utility = **1** then do;

absaccruals\_sub = **.**;

absda\_fe = **.**;

da\_fe = **.**;

end;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STEP 14. Export Subsidiary data to Stata dta file

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **export**

data=Caudit.orbis\_final\_allsubs

dbms=dta

outfile = 'FILEPATH\Subsidiary' replace;

**run**;