Contents:

1. Login and assign folder to Save dataset
2. [Import Compustat Annual Data](#Two)
3. [Merge Two Datasets](#THREE)
4. [Merge Three Datasets and Create One Lag and Two Lag Variables](#FOUR)
5. [Intx function to create date variables](#FIVE)
6. [No Duplicates](#SIX)
7. [Substring Function: Create date example](#SEVEN)
8. [Multiple Where’s or If’s](#EIGHT)
9. [Keep or Delete numbers in a range](#NINE)
10. [Regression with fixed effects and output of predicted value](#TEN)
11. [Proc Rank to create -ile variables](#ELEVEN)
12. [Proc SQL to Sum By a Group](#TWELVE)
13. [Create a Lead Variable](#THIRTEEN)
14. [Dechow Dichev Accruals](#FOURTEEN)
15. [Proc Freq to Count number of observation in a group](#FIFTEEN)
16. [Enumeration of observations by group 1,2,3,4,5 etc.](#SIXTEEN)
17. [Proc Survey Reg With test of significance of multiple variables](#SEVENTEEN)
18. [Import from Excel](#EIGHTEEN)
19. [Logistic Regression](#NINETEEN)
20. [Export file as Excel](#TWENTY)
21. [Export a STATA file](#TWENTYONE)

1. login

\*assign libname ‘saveData’ to the directory that follows;

libname saveData 'C:\Users\Aaron\Desktop\School\PHD\SAS\_Files';

\*permanently save dataset ‘data1’ as ‘maindata’ to the directory assigned above;

**data** savedata.maindata; set data1; **run**;

\*retrieve data from the directory assigned above and name it ‘Earnings’;

**data** Earnings; set savedata.maindata; **run**;

\*login prompt for wrds server;

%let wrds = wrds-cloud.wharton.upenn.edu 4016;

options comamid = TCP remote=wrds;

signon username = \_prompt\_;

2. Import Compustat

\* Pull Compustat data from wrds, create a new dataset called Compustat and download;

rsubmit;

\* Data step is used to create a new data file from another file/dataset;

**data** compustat; \* New file name is compann;

set comp.funda (keep= gvkey datadate cusip tic fyear fyr at

sale ib indfmt datafmt popsrc consol sich act che dlc dp lct ppent ppegt rect rectr lt oancf re ebit csho prcc\_F revt invt cik acqmeth );

if consol = 'C'; \*Consol = 'C' selects the firms consolidated financial statements;

if datafmt = 'STD'; \*Datafmt = 'STD' is restated data;

if indfmt = 'INDL'; \*indfmt = 'INDL' selects firms that report their data in industry format;

if popsrc = 'D'; \*popsrc = 'D' selects domestic firms (USA, Canada & ADRs);

if fyear GE **1999** and fyear LE **2014**; \*GE = Greater than or equal to, LE = Less than or equal to;

if gvkey = **.** then delete;

**run**; **quit**;

**proc** **download** data = compustat;

**quit**;

endrsubmit;

3. Merge two datasets

\*Merge datasets ‘Data1’ and ‘Data2’ to create dataset ‘Data3’, this proc sql statement retains all variables from dataset a (a.\*) and assets from dataset b (b.at, assumes asset variable name is ‘at’), the on part of the statement requires that firms have the same gvkey and fyear;

**proc** **sql**;

create table Data3

as select a.\*, b.at

from Data1 as a

left join Data2 as b

on a.gvkey = b.gvkey and a.fyear = b.fyear;

**quit**;

4. Merge three datasets and create one lag and two lag  
\*Create lagged variable and variable with two lags, Merge datasets ‘Data1’ and ‘Data2’ and ‘Data3’ to create dataset ‘Data4’, this proc sql statement retains all variables from dataset a (a.\*) and assets from datasets b and c (b.at, c.at, assumes asset variable name is ‘at’), the on part of the statement requires that firms have the same gvkey and fyear;

**proc** **sql**;

create table data4

as select a.\*, b.at as LagAssets1, c.at as LagAssets2

from data1 as a

left join data2 as b

on a.gvkey = b.gvkey and a.fyear = b.fyear+1

left join data3 as c

on a.gvkey = c.gvkey and a.fyear = c.fyear+2;

**quit**;

5. Intx function to create and manipulate dates

\*Use INTX function to create dates, could also use this function to create a new variable that is the same date of the prior month ex. Feb. 1 to Jan. 1;

**data** data2; set data1;

\*prioryear is the same date as datadate but one year earlier;

prioryear =intnx('year',datadate,-**1**, 'same');

format prioryear date10.;

\*fiscalstart is one day after prioryear;

fiscalstart =intnx('day', prioryear,+**1**);

format fiscalstart date10.;

\*nextyear is one year after datadate and makes sure to fall on the same day, not necessarily 365 days;

nextyear =intnx('year',datadate,+**1**, 'same');

format nextyear date10.;

**run**;

6. Delete Duplicates

\*creates new dataset ‘data2’ that only retains one observation for each firm (gvkey) and year (fyear);

**proc** **sort** data = data1 nodupkey out = data2;

by gvkey fyear;

**quit**;

7. Substring to break apart variable

\*creates new dataset ‘data2’ uses substrn function to pull parts of the variable into new variables, creates variable ‘year’ as the first four digits of filing\_date ‘1,4’ means starting at the first digit and taking 4 digits, ‘month’ pulls a two digit month from the variable ‘filing\_date’ the fifth and sixth digits ‘5,2’, and day pulls a two digit day variable from the seventh and eight digits of ‘filing\_date’, next I create a variable File\_date from the three variables I just created. This code was used to create a date variable from a date that SAS didn’t recognize, there is an easier way to do this but I use it as an example of the substrn function, the trim function can also be used for a similar purpose;

**data** Data2; set Data1;

format File\_date date10.;

year = substrn(filing\_date,**1**,**4**);

month = substrn(filing\_date,**5**,**2**);

day = substrn(filing\_date,**7**,**2**);

File\_date = mdy(month,day,year);

**run**;

8. Multiple where statement

\*Only keeps observations where the variable form is equal to 10-Q, 8-K, 4, DEF 14A, or 10-K;

**data** test2; set test1;

where form = '10-Q' or form = '8-K' or form ='4' or form ='DEF 14A' or form ='10-K';

**run**;

9. Keep or delete numbers in a range

\*Keep or delete numbers in a range;

**data** test2; set test1;

if sich in (**6000**:**6999**) then delete;

**run**;

10. Regression with Fixed Effects and Output Predicted Value

\*Regress log of compensation on independent variables with industry (sic2) and fyear fixed effects, predicted value for each firm observation is output as ‘expected\_comp’ into dataset ‘data2’ because of ‘p=expected\_comp’;

**proc** **surveyreg** data=data1;

class sic2 fyear;

model logcomp = logtenure logsales lagSNP500 LagBtoM Return LagReturn ROA LagROA sic2 fyear/solution;

output out = data2 p=expected\_comp;

**run**;

11. Proc Rank to create -ile variables

\*create quartile, quintile or any other -ile variables you desire, code creates dataset ‘data2’ with a new variable ‘decile\_comp’ that ranks the variable ‘ExcessComp’ into ten groups 0-9 (numbering always starts at 0, 0 being the lowest decile of ‘ExcessComp’;

**proc** **rank** data=data1 groups=**10** out=data2;

var ExcessComp;

ranks decile\_comp;

**run**;

12. Sum Variable by Group

\*Create dataset ‘Data2’ from ‘Data1’, and sum the variable ‘audit\_fees’ by fyear to create new variable YearFees;

**proc** **sql**;

create table Data2

as select distinct Fyear, sum(audit\_fees)as YearFees

from Data1

group by Fyear;

**quit**;

13. Lead variables

**proc** **sql**;

create table data3

as select a.\*, b.at as LeadAssets1

from data1 as a

left join data2 as b

on a.gvkey = b.gvkey and a.fyear = b.fyear-1;

**quit**;

14. Dechow Dichev Accruals

\*Dechow Dichev 2002 accruals measure (not my code);

**data** accruals; set Anderson7;

cyear = year(datadate);

keep gvkey fyear cyear sich act at che dlc dp lct ppent ppegt rect rectr sale ib industryff48;

**run**;

**data** accruals; set accruals;

if missing(SIC) then SIC = SICH;

**run**;

**proc** **sort** data = accruals nodupkey;

by gvkey fyear;

**run**;

\*create accruals variables;

**data** accruals; set accruals;

SIC2 = int(SIC/**100**);

\* Create lags;

GVKEYtm1 = lag(GVKEY);

FYEARtm1 = lag(FYEAR);

ACTtm1 = lag(ACT);

ATtm1 = lag(AT);

CHEtm1 = lag(CHE);

DLCtm1 = lag(DLC);

LCTtm1 = lag(LCT);

PPEGTtm1 = lag(PPEGT);

PPENTtm1 = lag(PPENT);

RECTtm1 = lag(RECT);

RECTRtm1 = lag(RECTR);

SALEtm1 = lag(SALE);

if GVKEY ne GVKEYtm1 OR FYEAR ne (FYEARtm1+**1**) then do;

ACTtm1 = **.**;

ATtm1 = **.**;

CHEtm1 = **.**;

DLCtm1 = **.**;

LCTtm1 = **.**;

PPEGTtm1 = **.**;

PPENTtm1 = **.**;

RECTtm1 = **.**;

RECTRtm1 = **.**;

SALEtm1 = **.**;

end;

\* Computation Variables;

ATtm1Inverse = **1** / ATtm1;

ChgSALEdATtm1 = (SALE-SALEtm1) / ATtm1;

ChgRECTdATtm1 = (RECT-RECTtm1) / ATtm1;

ChgSALEmChgRECTdATtm1 = ((SALE-SALEtm1)-(RECT-RECTtm1)) / ATtm1;

ChgSALEmChgRECTRdATtm1 = ((SALE-SALEtm1)-(RECTR-RECTRtm1)) / ATtm1;

PPEGTdATtm1 = PPEGT/ATtm1;

ROAdATtm1 = IB/ATtm1;

\* Calculate Current Accruals as in Dechow, Sloan, and Sweeney (1995);

CADechowEtAl = (ACT-ACTtm1)-(LCT-LCTtm1)

-(CHE-CHEtm1)+(DLC-DLCtm1)-DP;

CAdTAtm1DechowEtAl = CADechowEtAl/ATtm1;

**run**;

\* Prepare Estimation of Non-Discretionary Accruals;

**data** EstimationInput; set accruals;

KEEP CYEAR industryff48 CAdTAtm1DechowEtAl ATtm1Inverse

ChgSALEdATtm1 ChgSALEmChgRECTdATtm1 PPEGTdATtm1 ROAdATtm1;

**run**;

**proc** **sort** data = EstimationInput;

by CYEAR industryff48;

**run**;

**proc** **reg** data = EstimationInput

noprint tableout edf outest = ModJones1991EstimationInt;

by CYEAR industryff48;

model CAdTAtm1DechowEtAl = ATtm1Inverse ChgSALEmChgRECTdATtm1

PPEGTdATtm1 ROAdATtm1;

**quit**;

**data** ModJones1991EstimationInt; set ModJones1991EstimationInt;

where \_TYPE\_ = 'PARMS';

ModJones1991Intercept = Intercept;

ModJones1991IntATParm = ATtm1Inverse;

ModJones1991IntSALERECParm = ChgSALEmChgRECTdATtm1;

ModJones1991IntPPEParm = PPEGTdATtm1;

ModJones1991IntROAParm = ROAdATtm1;

if \_P\_ < **5** then do;

ModJones1991Intercept = **.**;

ModJones1991IntATParm = **.**;

ModJones1991IntSALERECParm = **.**;

ModJones1991IntPPEParm = **.**;

ModJones1991IntROAParm = **.**;

end;

LABEL ModJones1991Intercept = 'Modified Jones (1991) Model w/ Intercept: Intercept Parameter'

ModJones1991IntATParm = 'Modified Jones (1991) Model w/ Intercept: Inverse of Assets Parameter'

ModJones1991IntSALERECParm = 'Modified Jones (1991) Model w/ Intercept: Chg Sales less Chg Rec Parameter'

ModJones1991IntPPEParm = 'Modified Jones (1991) Model w/ Intercept: PP&E Parameter'

ModJones1991IntROAParm = 'Modified Jones (1991) Model w/ Intercept: ROA Parameter';

KEEP CYEAR industryff48

ModJones1991Intercept ModJones1991IntATParm

ModJones1991IntSALERECParm ModJones1991IntPPEParm

ModJones1991INTROAParm;

**run**;

/\* Combine estimations for importing and to save for archival purposes \*/

**data** EarningsManagementEstimations; set EstimationInput;

KEEP industryff48 CYEAR;

LABEL industryff48 = 'FamaFrench'

CYEAR = 'Calendar Year';

**run**;

**proc** **sort** data = EarningsManagementEstimations nodupkey;

by CYEAR industryff48;

**run**;

**proc** **sql**;

create table EarningsManagementEstimations1 as

select a.\*,

b.ModJones1991Intercept, b.ModJones1991IntATParm,

b.ModJones1991IntSALERECParm, b.ModJones1991IntPPEParm,

b.ModJones1991INTROAParm

from EarningsManagementEstimations as a LEFT JOIN ModJones1991EstimationInt as b

on a.CYEAR = b.CYEAR and

a.industryff48 = b.industryff48;

**quit**;

\*use estimations to compute discretionary accruals;

**proc** **sql**;

create table EarningsManagement as

select a.\*,

b.ModJones1991Intercept, b.ModJones1991IntATParm,

b.ModJones1991IntSALERECParm, b.ModJones1991IntPPEParm,

b.ModJones1991INTROAParm

from accruals as a LEFT JOIN EarningsManagementEstimations1 as b

on a.CYEAR = b.CYEAR and

a.industryff48 = b.industryff48;

**quit**;

**data** EarningsManagementFinal; set EarningsManagement;

TCAModJones1991 = CAdTAtm1DechowEtAl;

NDCAModJones1991Int = ModJones1991Intercept

+ (ModJones1991IntATParm \* ATtm1Inverse)

+ (ModJones1991IntSALERECParm \* ChgSALEmChgRECTdATtm1)

+ (ModJones1991IntPPEParm \* PPEGTdATtm1)

+ (ModJones1991IntROAParm \* ROAdATtm1);

DCAModJones1991Int = TCAModJones1991 - NDCAModJones1991Int;

LABEL CAdTAtm1DechowEtAl = 'Total Current Accruals - Dechow, Sloan, and Sweeney (1995)'

NDCAModJones1991Int = 'Nondiscretionary Current Accruals - Modified Jones (1991) Model w/ Intercept'

DCAModJones1991Int = 'Discretionary Current Accruals - Modified Jones (1991) Model w/ Intercept';

KEEP GVKEY FYEAR

CAdTAtm1DechowEtAl NDCAModJones1991Int DCAModJones1991Int;

**run**;

**data** EarningsManagementFinal;

RETAIN GVKEY FYEAR

CAdTAtm1DechowEtAl

NDCAModJones1991Int DCAModJones1991Int;

set EarningsManagementFinal;

**run**;

**proc** **sort** data = EarningsManagementFinal nodupkey;

by GVKEY FYEAR;

**run**;

\*add accruals information;

\*need to put other table instead of restate;

**proc** **sql**;

create table contaminated

as select a.\*, b.CAdTAtm1DechowEtAl, b.NDCAModJones1991Int, b.DCAModJones1991Int

from Anderson7 as a

right join earningsmanagementfinal as b

on a.gvkey=b.gvkey and a.fyear = b.fyear;

**quit**;

15. Proc Freq to count the number of observations in a group

\*provides a count of the number of observations that have the same gvkey auditor pairing;

**proc** **freq** data = consecutive noprint;

tables gvkey\*auditor\_fkey/out = consecutive1 list nopct nocum;

**run**;**quit**;

16. Enumeration of Observations

\*numbers observations that have the same cik and year;

**data** data2; set data1;

count +**1**;

by cik year;

if first.year then count =**1**;

**run**;

17. Testing significance of more than one variable

**proc** **surveyreg** data = feeregchangeswin;

cluster gvkey;

class fyear industryff48;

model chgfees = lagRelCon LagBigRelcon chgAT chg02 chglev/solution ;

estimate lagRelCon **1** LagBigRelcon **1**; \*this tests if these two added are significant;

**run**;

18. Import Excel

\*may need to replace ‘xlsx’ with ‘csv’ or ‘xls’ based on file type, creates file named ‘Data1’ from excel file ‘Master’, datarow = 2 acknowledges the fact that the first row has variable names;

**PROC** **IMPORT** OUT= WORK.Data1

DATAFILE= "C:\PHD\Research\StockPledged\Master.xlsx"

DBMS=xlsx REPLACE;

GETNAMES=YES;

DATAROW=**2**;

**RUN**;

19. Logistic Regression

**proc** **logistic** data = Comp7bCol1 descending;

model repurchindicator = ROA PastReturn Cash / rsq;

**run**;

20. Export Excel

**proc** **export** data=Clean5 outfile='C:\Stock Pledged as Collateral\BasicPledge.xlsx'

dbms = xlsx replace;

**run**;

21. Export Stata

**proc** **export** data=contaminatedpca outfile= "C:\Users\Aaron\Desktop\School\PHD\SAS\_Files\contaminatedpca.dta";

**run**;