**Dimas’ Replicatication of HL 2005 JBFA**

**1.- Sas – Compustat Crsp Universe**

/\*

AUTHOR: F. Dimas Pena Romera

START DATE: 27/12/14

LAST MODIFIED: 08/01/15

PURPOSE: Download selected COMPUSTAT VARIABLES from Compustat annual file, merge with crsp permno using wrds linkfile and

for selected YEARS, merge with CRSP using merge link and SELECTED LINKS, spit Compustat-Crsp Universe

INPUT:

1. location of raw compustat and linkfile after rawcomp.

2. location of destination merged file after dimmod.

3. location of descriptives (number of unique gvkeys linked per year in a csv file).

4. beg fyear and end fyear after.

5. variables requested from compustat annual.

OUTPUT:

1. dimmod.filename = compustat-crsp merged file with requested compustat variables.

\*/

\*include macros;

%include 'C:\Users\penarome\Desktop\Academic\UNCPP2HL\Scripts\macros.sas';

\*I define my local permanent libraries in which I will modify and update outputs.;

libname dimmod 'C:\Users\penarome\Desktop\Academic\UNCPP2HL\modified' ;

libname fig 'C:\Users\penarome\Desktop\Academic\UNCPP2HL\HLWriting';

\*I define libraries in which to locate and extract rawdata. all these directories will be deleted at the end of the project;

libname rawcomp 'C:\Users\penarome\Desktop\Academic\RAW DATABASES\RCompustat\_2015' ;

\* get required data from compustat annual;

\* comp.fundq (if quarterly data, change variables as well, just add q to the variable);

\* Date range-- applied to FYEAR (Fiscal Year);

%let fyear1= 1984;

%let fyear2= 2014;

\* Selected data items (GVKEY, DATADATE, FYEAR and FYR are automatialy included);

\*I also require;

%let vars=

substr(CUSIP,1, 8) as CUSIP,

fyear,

conm,

gvkey,

datadate,

fyr,

seq,

PRCC\_C AS pxfye ,

CSHO AS shsfye ,

NI AS ni ,

IBCOM AS ibxd ,

SSTK AS cpsale ,

PRSTKC AS cppur ,

DVC AS div ,

CEQ AS ce ,

AT AS ta ,

XINT AS intexp ,

XRD AS rd ,

IB AS ibx ,

XINT AS junk ,

TXDI AS isdt ,

ITCI AS isitc ,

TXT AS itx ,

CSHPRI AS shseps ,

PSTKRV AS pfdred ,

PSTKL AS pfdliq ,

PSTK AS pfdpar ,

LT AS tl ,

TXDITC AS bsdt ,

SPI AS spec,

CAPX;

**proc** **sql** ;

create table compa as

select distinct &vars

from rawcomp.R\_2015\_funda

where (fyear between &fyear1 and &fyear2) & (Consol='C') & (Datafmt='STD' and Popsrc='D' and Indfmt= 'INDL')

order by gvkey, datadate;

**quit**;

\*get sic codes from names file;

**proc** **sql** ;

create table compa as

select distinct A. \*, B.SIC as dnum

from compa A, rawcomp.R\_2015\_names B

where (A.GVKEY=B.GVKEY)

order by gvkey, datadate;

**quit**;

\*delete duplicates by gvkey datadate (none found);

**proc** **sort** data =compa nodupkey;

by gvkey datadate;

**run**;

\*I generate variables for the beginning and end of fiscal year (these will be used to link with crsp);

**data** compa;

set compa;

format endfyr begfyr date9.;

endfyr=datadate;

begfyr= intnx('month',endfyr,-**11**,'beg');

**run**;

\*I get the compu crsp merge table

\*My compustat-crsp universe will be defined as the set of compustat firms with a valid link in the linktable. I limit linktypes to LU LC and LS (this captures

most compustat-crsp links without duplicated entries). Also, compustat datadate (end of fiscal year) is required to be between the valid link ranges in linktable (more

linking options in the square below);

**proc** **sql**;

create table compa as select distinct

a.\*, b.lpermno as permno, b.linktype, b.linkprim, b.liid, b.usedflag, b.LINKDT, b.LINKENDDT

from compa as a, rawcomp.R\_2015\_ccmxpf\_linktable as b

where (a.gvkey = b.gvkey)

and b.linktype in ('LU', 'LC', 'LS')

and (b.LINKDT <= a.endfyr or b.LINKDT = **.B**)

and (a.endfyr <= b.LINKENDDT or b.LINKENDDT = **.E**);

**quit**;

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

\* The previous condition requires the end of fiscal year to fall within the link range. \*

\* \*

\* A more relaxed condition would require any part of the fiscal year to be within the link range: \*

\* (b.LINKDT <= a.endfyr or missing(b.LINKDT) = 1) and (b.LINKENDDT >= a.begfyr or missing(b.LINKENDDT)= 1);\*

\* or a more strict condition would require the entire fiscal year to be within the link range : \*

\* (b.LINKDT <= a.begfyr or missing(b.LINKDT) = 1) and (a.endfyr <= b.LINKENDDT or b.LINKENDDT= .E) \*

\* \*

\* If these conditions are used, we suggest using the result data set from the "collapsing" procedure - \*

\* which is shown in sample program ccm\_lnktable.sas - to replace crsp.ccmxpf\_linktable. \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**data** compa; set compa;

if missing(permno)=**0**;

**run**;

\*no gvkey-permno-datadate duplicates

\*!!!!!!!!I notice there are still some duplicated gvkey-datadate combinations. I sort firms on gvkey datadate liid and then drop duplicates.

This keeps the observation with the lowest liid value.;

**proc** **sort** data=compa out=compa nodupkey;

by gvkey datadate permno;

**run**;

**proc** **sort** data=compa out=compa;

by gvkey datadate liid;

**run**;

**proc** **sort** data=compa out=compa nodupkey;

by gvkey datadate;

**run**;

\*I generate dimmod.compmerged as our Compustat Crsp Univ;

**proc** **sort** data=compa out=dimmod.compcrsp;

by gvkey datadate;

**run**;

\*I generate a sata file dimmod.comperged as my Compustat Crsp Universe;

**proc** **export**

data= dimmod.compcrsp

dbms=dta

outfile="C:\Users\penarome\Desktop\Academic\UNCPP2HL\modified\compcrsp.dta"

replace;

**run**;

\*clear temporary libraries;

libname rawcomp clear;

/\*just to contrast whether my compustat-crsp universe is consistent with prior literature, I look at how many distinct gvkeys I get per year

(numbers are very reasonable - minor differences with http://gridgreed.blogspot.com.es/2012/12/on-merging-crsp-and-compustat-data.html)

I export to csv to get tables in excel;

proc sort data=dimmod.compcrsp out=dimmod.compcrsp;

by fyear;

run;

proc sql;

create table fig.yearobs

as select a.fyear, n(gvkey) as n

from dimmod.compcrsp a

group by fyear;

quit;

proc export data=fig.yearobs (where=(fyear ge 1981))

outfile='C:\Users\penarome\Desktop\Academic\UNCPP2HL\HLWriting\yearobs.csv'

dbms=csv

replace;

run;

**2.- Sas – Cleaning compustat data for basic table of HL**

/\*

AUTHOR: F. Dimas Pena Romera

START DATE: 27/12/14

LAST MODIFIED: 08/01/15

PURPOSE: Prepare compustat data for analysis in stata. Ohlson model and descriptives

INPUT: dimmod.compucrsp was created in a\_Compustat\_Crsp\_Universe script with the required data for the following analysis.

OUTPUT: compcrsp.dta (stata file - ready for analysis)

\*/

\*merge compustat crsp universe with variables from ibes\_permno, note that ibes\_permno has duplicates that

we must deal with;

/\*

proc sql;

create table ibescompu as select a.\*, b.actual, b.anndats\_act, b.sur, b.cusip, b.shout as ibesshout, b.price as ibesprice from dimmod.compcrsp a left join dimmod.sum b

on a.permno=b.permno and a.datadate=b.fpedats

where a.permno ne .;

quit;

\*/

\*logical checks

\* ------------------------------------------------------------------- \*;

\* Create basic variables.

\* ------------------------------------------------------------------- \*;

\* Logical checks and replacing of missing with zero and vice-versa,

\* after having looked at raw data.

\*------------------------------------------------------------------- \*;

**data** compucrsp; set dimmod.compcrsp;

if pxfye <= **0** then pxfye = **.**; \*if price at end of fiscal year is negative then set to missing.;

if shsfye <= **0** then shsfye = **.**; \*if shares outstanding and the end of fiscal year is negative then set to missing.;

if div < **0** then div = **.**; \*if dividends negative then set to missing;

if ta <= **0** then ta = **.**; \* if total assets negative then set to missing;

if tl <= **0** then tl = **.**; \*if total liabilities negative then set to missing;

if rd = **.** then rd = **0**; \*if R&D exp is missing then assume its 0 (alternatively a dummy could be used);

if ce = **0** then ce = **.**; \*if common equity is 0 then set to missing (distressed companies?);

**run**;

\*Industries;

\* ------------------------------------------------------------------- \*;

\* Define industries by BBL98 p.43 classifications.

\* 1. sic = 1 Mining and construction

\* 2. sic = 2 Food

\* 3. sic = 3 Textiles, printing and publishing

\* 4. sic = 4 Chemicals

\* 5. sic = 5 Pharmaceuticals

\* 6. sic = 6 Extractive industries

\* 7. sic = 7 Durable manufacturers

\* 8. sic = 8 Computers

\* 9. sic = 9 Transportation

\* 10. sic = 10 Utilities

\* 11. sic = 11 Retail

\* 12. sic = 12 Financial institutions

\* 13. sic = 13 Insurance and real estate

\* 14. sic = 14 Services

\* ------------------------------------------------------------------- \*;

**data** compucrsp; set compucrsp;

sic = **0**;

if dnum >= **1000** and dnum <= **1999** then sic = **1**;

if dnum >= **2000** and dnum <= **2111** then sic = **2**;

if dnum >= **2200** and dnum <= **2780** then sic = **3**;

if dnum >= **2800** and dnum <= **2824** then sic = **4**;

if dnum >= **2840** and dnum <= **2899** then sic = **4**;

if dnum >= **2830** and dnum <= **2836** then sic = **5**;

if dnum >= **2900** and dnum <= **2999** then sic = **6**;

if dnum >= **1300** and dnum <= **1399** then sic = **6**;

if dnum >= **3000** and dnum <= **3999** then sic = **7**;

if dnum >= **3580** and dnum <= **3579** then sic = **8**;

if dnum >= **3670** and dnum <= **3679** then sic = **8**;

if dnum >= **7370** and dnum <= **7379** then sic = **8**;

if dnum >= **4000** and dnum <= **4899** then sic = **9**;

if dnum >= **4900** and dnum <= **4999** then sic = **10**;

if dnum >= **5000** and dnum <= **5999** then sic = **11**;

if dnum >= **6000** and dnum <= **6411** then sic = **12**;

if dnum >= **6500** and dnum <= **6999** then sic = **13**;

if dnum >= **7000** and dnum <= **7369** then sic = **14**;

if dnum >= **7380** and dnum <= **8999** then sic = **14**;

if sic = **1** then dsic1 = **1**; else dsic1 = **0**;

if sic = **2** then dsic2 = **1**; else dsic2 = **0**;

if sic = **3** then dsic3 = **1**; else dsic3 = **0**;

if sic = **4** then dsic4 = **1**; else dsic4 = **0**;

if sic = **5** then dsic5 = **1**; else dsic5 = **0**;

if sic = **6** then dsic6 = **1**; else dsic6 = **0**;

if sic = **7** then dsic7 = **1**; else dsic7 = **0**;

if sic = **8** then dsic8 = **1**; else dsic8 = **0**;

if sic = **9** then dsic9 = **1**; else dsic9 = **0**;

if sic = **10** then dsic10 = **1**; else dsic10 = **0**;

if sic = **11** then dsic11 = **1**; else dsic11 = **0**;

if sic = **12** then dsic12 = **1**; else dsic12 = **0**;

if sic = **13** then dsic13 = **1**; else dsic13 = **0**;

if sic = **14** then dsic14 = **1**; else dsic14 = **0**;

**run**;

\* ------------------------------------------------------------------- \*;

\* OHLSON-specific variables.

\* ------------------------------------------------------------------- \*;

**data** compucrsp; set compucrsp;

netcap = cppur - cpsale;

ione = ni - ibx ;

mve = pxfye\*shsfye;

ionepct = **.**;

d\_ione = **0**;

if ibx ne **0** then ionepct = abs(ione)/abs(ibx);

if ione < **0** or ione > **0** then d\_ione = **1**;

if ione = **.** then d\_ione = **.**;

**run**;

\*I generate a sata file dimmod.comperged as my Compustat Crsp Universe;

**proc** **export**

data= compucrsp

dbms=dta

outfile="C:\Users\penarome\Desktop\Academic\UNCPP2HL\modified\compcrsp.dta"

replace;

**run**;

**3. Stata – Finish cleaning and run analysis**

clear all

cd "C:\Users\penarome\Desktop\Academic\UNCPP2HL\modified"

use compcrsp.dta, clear

cd "C:\Users\penarome\Desktop\Academic\UNCPP2HL\HLWriting"

\*note that because of the construction of the compsutat crsp universe we have no

\*duplicates in gvkey fyear

duplicates drop gvkey fyear, force

destring(gvkey) , replace

\*set data set as pannel to create lags and leads

xtset gvkey fyear, yearly

\* \* ------------------------------------------------------------------- \*;

\* Relevant leads and lags

\* --------------------------------------------------------------------- \*;

gen cem1 = l.ce

gen mvem1=l.mve

gen ibxm1=l.ibx

gen divm1=l.div

\* ------------------------------------------------------------------- \*;

\* Define dirty surplus items DIRT.

\* ------------------------------------------------------------------- \*;

browse gvkey fyear ce ibx ni netcap div

gen dirt = ce - cem1 - ni + div + netcap

gen d\_dirt=0

gen dirtpct=.

replace dirtpct=abs(dirt)/abs(ibx) if ibx!=0

replace d\_dirt=1 if abs(dirt) > 0.01

replace d\_dirt=. if dirt==.

\* ------------------------------------------------------------------- \*;

\* Define abnormal earnings.

\* ------------------------------------------------------------------- \*;

gen r = 0.12

gen xa= ibx - (r\*l.ce)

gen xad1=f.xa

gen xam1=l.xa

\* ------------------------------------------------------------------- \*;

\* Create various dummies

\* ------------------------------------------------------------------- \*;

gen d\_div = 1 if div>0

replace d\_div=0 if div==0

replace d\_div=. if div==.

assert div>=0

browse gvkey fyear ce

gen d\_nce = 1 if ce<=0

replace d\_nce=0 if d\_nce!=1

replace d\_nce=. if ce==.

browse gvkey fyear ibx

gen d\_nibx = 1 if ibx<=0

replace d\_nibx=0 if d\_nibx!=1

replace d\_nibx=. if ibx==.

\* ------------------------------------------------------------------- \*;

\* keep relevant variables

\* ------------------------------------------------------------------- \*;

keep gvkey fyear fyr conm pxfye shsfye mve mvem1 ce cem1 ibx ibxm1 ni ///

div divm1 netcap rd ta tl xam1 xa xad1 dirt dirtpct d\_dirt ione ionepct ///

d\_ione d\_div d\_nce d\_nibx sic capx ///

dsic1 dsic2 dsic3 dsic4 dsic5 dsic6 dsic7 dsic8 dsic9 dsic10 ///

dsic11 dsic12 dsic13 dsic14

\* ------------------------------------------------------------------- \*;

\* Delete if missing mve, ce, ibx, div or netcap.

\* ------------------------------------------------------------------- \*;

drop if fyear ==.

drop if sic ==0

drop if sic ==.

drop if mve ==.

drop if ce ==.

drop if ibx ==.

drop if netcap ==.

drop if div ==0

\* ------------------------------------------------------------------- \*;

\* Winsorize / Trim variables (Following HL2005 I trim by year at 1 and 99 pct)

\* ------------------------------------------------------------------- \*;

foreach x in mve ce ibx div netcap xam xa xad1 rd {

winsor2 `x' , replace cuts(1 99) trim by(fyear)

}

\* ------------------------------------------------------------------- \*;

\* Delete again after trimming

\* ------------------------------------------------------------------- \*;

drop if fyear ==.

drop if sic ==0

drop if sic ==.

drop if mve ==.

drop if ce ==.

drop if ibx ==.

drop if netcap ==.

drop if div ==0

\* ------------------------------------------------------------------- \*;

\* Descriptive evolution of payout policy

\* ------------------------------------------------------------------- \*;

browse gvkey fyear netcap div ni ibx ta

gen netcap\_ni= netcap/ni if ni>0 & ni !=.

gen netcap\_ta= netcap/ta if ta>0 & ta !=.

gen div\_ni = div/ni

gen payout=(netcap + div)/ni

\* ------------------------------------------------------------------- \*;

\* Run Ohlson by year and store coefficients

\* ------------------------------------------------------------------- \*;

gen c1=.

gen c2=.

gen c3=.

gen c4=.

gen Rsq=.

/\*

gen Vc1=.

gen Vc2=.

gen Vc3=.

gen Vc4=.

\*/

su fyear

local xmin r(min)

local xmax r(max)

scalar xmin = r(min)

scalar xmax = r(max)

forv fy = 1984(1)2014 {

reg mve ce ibx div netcap i.sic if fyear==`fy'

mat b\_`fy'=e(b)

\*mat li b\_1998

replace c1=b\_`fy'[1,1] if fyear==`fy'

replace c2=b\_`fy'[1,2] if fyear==`fy'

replace c3=b\_`fy'[1,3] if fyear==`fy'

replace c4=b\_`fy'[1,4] if fyear==`fy'

replace Rsq=e(r2) if fyear==`fy'

/\*

mat V\_`fy'=e(V)

\* mat li V\_2014

replace Vc1=sqrt(V\_`fy'[1,1]) if fyear==`fy'

replace Vc2=sqrt(V\_`fy'[2,2]) if fyear==`fy'

replace Vc3=sqrt(V\_`fy'[3,3]) if fyear==`fy'

replace Vc4=sqrt(V\_`fy'[4,4]) if fyear==`fy'

\*/

}

preserve

duplicates drop fyear, force

drop if fyear > 1995

keep fyear c1 c2 c3 c4 Rsq

su c1 c2 c3 c4 Rsq

gen c3\_c1=c3-c1

gen c4\_c1=c4-c1

outsheet fyear c1 c2 c3 c4 Rsq c3\_c1 c4\_c1 using 84to95.csv , comma replace

restore

preserve

duplicates drop fyear, force

drop if fyear < 1996

su c1 c2 c3 c4 Rsq

gen c3\_c1=c3-c1

gen c4\_c1=c4-c1

outsheet fyear c1 c2 c3 c4 Rsq c3\_c1 c4\_c1 using 96to14.csv , comma replace

restore

/\* Alternatively I could run the tests directly from stata - I check some

examples and excel is performing the same tests.

drop if fyear>1995

duplicates drop fyear, force

keep fyear c1 c2 c3 c4 Rsq

su c1 c2 c3 c4 Rsq

gen c3\_c1=c3-c1

gen c4\_c1=c4-c1

ttest c1==0

ttest c4==0

ttest c4\_c1==-1

ttest c1==0

One-sample t test

------------------------------------------------------------------------------

Variable | Obs Mean Std. Err. Std. Dev. [95% Conf. Interval]

---------+--------------------------------------------------------------------

c1 | 12 .6439595 .0509785 .1765946 .5317566 .7561623

------------------------------------------------------------------------------

mean = mean(c1) t = 12.6320

Ho: mean = 0 degrees of freedom = 11

Ha: mean < 0 Ha: mean != 0 Ha: mean > 0

Pr(T < t) = 1.0000 Pr(|T| > |t|) = 0.0000 Pr(T > t) = 0.0000

.

end of do-file

. do "C:\Users\penarome\AppData\Local\Temp\STD00000000.tmp"

. ttest c4==0

One-sample t test

------------------------------------------------------------------------------

Variable | Obs Mean Std. Err. Std. Dev. [95% Conf. Interval]

---------+--------------------------------------------------------------------

c4 | 12 .3073569 .267129 .9253619 -.2805901 .8953038

------------------------------------------------------------------------------

mean = mean(c4) t = 1.1506

Ho: mean = 0 degrees of freedom = 11

Ha: mean < 0 Ha: mean != 0 Ha: mean > 0

Pr(T < t) = 0.8629 Pr(|T| > |t|) = 0.2743 Pr(T > t) = 0.1371

.

end of do-file

. do "C:\Users\penarome\AppData\Local\Temp\STD00000000.tmp"

. ttest c4\_c1==-1

One-sample t test

------------------------------------------------------------------------------

Variable | Obs Mean Std. Err. Std. Dev. [95% Conf. Interval]

---------+--------------------------------------------------------------------

c4\_c1 | 12 -.3366026 .2688866 .9314504 -.928418 .2552127

------------------------------------------------------------------------------

mean = mean(c4\_c1) t = 2.4672

Ho: mean = -1 degrees of freedom = 11

Ha: mean < -1 Ha: mean != -1 Ha: mean > -1

Pr(T < t) = 0.9844 Pr(|T| > |t|) = 0.0313 Pr(T > t) = 0.0156

/\*