```
* Author: Jordan Schoenfeld;
* "Contracts between Firms and Shareholders";
* Journal of Accounting Research;
* Description: Merge in additional data from CRSP, Compustat, Factset, I/B/E/S, and the SEC. Then output dataset to Stata.;

libname rawd 'C:\data\shareholder_contracts';
libname homed 'C:\data\shareholder_contracts';

*****;
* Import results.txt from the Perl output in the previous data step using SAS Enterprise.;
*****;

* Clean up data from Perl.;
data results2;
    set rawd.results;
    id = _N_;
    * if missing(target_cik) then delete;
    cik_num = input(target_cik,12.);
    cik_char = left(put(target_cik,$12.));
    * Later we will Merge in CRSP-Compustat file by FILENAME from EDGAR and Compustat, so clean up the filename variable.;
    * As of now, the file name is its full file name as stored on the local directory.;
    * The next steps remove all excess notation from the filename so that it comports with Compustat.;
    new_filename = filename;
    new_filename = tranwrd(new_filename, "_1.txt", "");
    new_filename = tranwrd(new_filename, "_2.txt", "");
    new_filename = tranwrd(new_filename, "_3.txt", "");
    new_filename = tranwrd(new_filename, "_4.txt", "");
    new_filename = tranwrd(new_filename, ".txt", "");
    new_filename = tranwrd(new_filename, ".tx", "");
    new_filename = tranwrd(new_filename, ".t", "");
    * Get last digits for the file name (by not including a third parameter it grabs the remainder of expression). This will link to SEC Analytics Suite.;
    filename2 = substr(new_filename, length(new_filename)-19);
run;

* Remove any duplicate filenames.;
proc sort data=results2 out=resultsnodup nodupkey;
    by filename2;
run;

* Start with SEC Analytics Suite as a base.;
data secanalytics;
    set homed.SEC_Analytics_Jordan;
    * CIK in Compustat is numeric. Make it a char so it can be matched with SeekEdgar.;
    cik_num = input(cik,12.);
    cik_char = left(put(cik_num,$12.));
    * Only keep 13-D;
    if form ^= "SC 13D" then delete;
    * Get last digits for the file name since we will match by filename with the Perl-based dataset.;
    new_filename = tranwrd(SECFNAME, ".txt", "");
    filename2 = substr(new_filename, length(new_filename)-19);
    year = year(fdate);
run;

* Merge perl results to SEC analytics by the filename.;
proc sql;
    create table master_merged as
    select distinct a.*, b.*
    from secanalytics as a
    left join results2 as b
    on (a.filename2 = b.filename2);
```

```
quit;

data master_merged_2;
    set master_merged;
    raw_year = year(fdate);
run;

* Prepare Compustat for merge.;
data compustat;
    set homed.compcrsp_merged_jan2019;
    year = year(datadate);
    cik_num = input(cik,12.);
    cik_char = left(put(cik_num,$12.));
    comp_fdate = fdate;
    drop fdate;
run;

* Merge in Compustat by CIK/year.;
proc sql;
    create table master_merged3 as
    select distinct a.*, b.*
    from master_merged_2 as a
    left join compustat as b
    on (a.cik = b.cik) and (a.year = b.year);
quit;

* Check data;
proc means data = master_merged3 nway;
    class raw_year;
    var at;
run;

* Delete any obs with missing Compustat data.;
data master_merged4;
    set master_merged3;
    if missing(gvkey) = 1 then delete;
run;

* More checks of the data and removal of any duplicate obs.;
proc sort data=master_merged4;
    by filename2 DESCENDING gvkey;
run;

proc sort data=master_merged4 out=master_merged5 nodupkey;
    by filename2 gvkey;
run;

proc sort data=master_merged5 out=master_merged6 nodupkey;
    by secfname;
run;

proc sort data=master_merged6;
    by id descending gvkey;
run;

proc sort data=master_merged6 out=master_merged7 nodupkey;
    by id;
run;

proc means data = master_merged7 nway;
    class raw_year;
    var at fincf;
run;

proc univariate data=master_merged7 freq;
   var raw_year;
```

```
run;

* Create new variables.;
data master_merged8;
    set master_merged7;
    sic2 = int(sic/100);
    * Create filedate yearq;
    filedate_yearq = (year(fdate)*10)+qtr(fdate);
    * Create 6-digit cusip;
    cusip6 = substr(cusip, 1, 6);
run;

* More checks of the data.;
proc univariate data=master_merged8 freq;
   var filedate_yearq;
run;

proc univariate data=master_merged8 freq;
   var filedate_yearq;
run;

* Get market return and other data directly through WRDS;
data for_crsp;
    set master_merged8(keep = id fdate lpermno sic2);
run;

%let wrds = wrds-cloud.wharton.upenn.edu 4016;
options comamid=TCP remote=WRDS;
signon username=_prompt_;

rsubmit;

proc upload data = for_crsp out = work.for_crsp; quit;

* Get firm measures from [-5, +5] around filedate;
proc sql;
    create table for_crsp_firm_annc_pm5 as
    select a.id,
        EXP(sum(log(1+b.RET)))-1 as firmreturns_pm5,
        AVG((ask-bid)/((ask+bid)/2)) as firmspread_pm5,
        AVG(abs(prc)*abs(shrout*1000)) as firmvalue_pm5,
        AVG((abs(ret)/abs(vol*prc))*1000000) as firmamihud_pm5,
        AVG(abs(vol)) as firmvolume_pm5,
        AVG(abs(vol)/abs(shrout*1000)) as firmturnover_pm5,
        SQRT(VAR(b.RET)) as firmretsd_pm5
    from for_crsp as a
    left join crsp.dsf as b
    on (a.lpermno = b.permno and b.date <= intnx('weekday',a.fdate,+5) and b.date >=
    intnx('weekday',a.fdate,-5))
    group by a.id;
quit;

* Get firm measures from [-365, -5] around filedate;
proc sql;
    create table for_crsp_firm_annc_m365 as
    select a.id,
        EXP(sum(log(1+b.RET)))-1 as firmreturns_m365,
        AVG((ask-bid)/((ask+bid)/2)) as firmspread_m365,
        AVG(abs(prc)*abs(shrout*1000)) as firmvalue_m365,
        AVG((abs(ret)/abs(vol*prc))*1000000) as firmamihud_m365,
        AVG(abs(vol)) as firmvolume_m365,
        AVG(abs(vol)/abs(shrout*1000)) as firmturnover_m365,
        SQRT(VAR(b.RET)) as firmretsd_m365
    from for_crsp as a
    left join crsp.dsf as b
    on (a.lpermno = b.permno and b.date < intnx('weekday',a.fdate,-5) and b.date >=
```

```
    intnx('day',a.fdate,-365))
    group by a.id;
quit;

* Returns data sets (market return benchmarks are created and downloaded further below);
proc download data = for_crsp_firm_annc_pm5 out = for_crsp_firm_annc_pm5; quit;
proc download data = for_crsp_firm_annc_m365 out = for_crsp_firm_annc_m365; quit;

* Get firm-level institutional holdings data;
data s34_2;
    set tfn.s34 (keep = cusip rdate shares shrout1);
    if year(rdate) < 1993 then delete;
    if missing(cusip) then delete;
    if missing(shrout1) then delete;
    if shrout1 = 0 then shrout1 = 1000000;
    if missing(shares) then delete;
    shares_pct = shares/(shrout1*1000000);
    cusip6 = substr(cusip, 1, 6);
    yearq = (year(rdate)*10)+qtr(rdate);
run;

proc sql;
    create table iiholdings as
    select cusip6, yearq, sum(shares_pct) as ii_shares_pct
    from s34_2
    group by cusip6, yearq;
quit;

proc download data = iiholdings out = homed.iiholdings; quit;
* End firm-level institutional holdings data;

* Get value-weighted returns;
proc sql;
    create table for_crsp_vwew_annc_pm5 as
    select a.id,
        EXP(sum(log(1+b.VWRETD)))-1 as vwreturns_filedate_pm5,
        EXP(sum(log(1+b.EWRETD)))-1 as ewreturns_filedate_pm5
    from for_crsp as a
    left join crsp.dsi as b
    on (b.date <= intnx('weekday',a.fdate,+5) and b.date >= intnx('weekday',a.fdate,-5))
    group by a.id;
quit;

proc download data = for_crsp_vwew_annc_pm5 out = for_crsp_vwew_annc_pm5; quit;

* Sign off WRDS;
endrsubmit;
signoff;

* Check mean institutional holdings;
proc means data = homed.iiholdings; run;

* Sort before merging;
proc sort data = for_crsp_firm_annc_pm5; by id; run;
proc sort data = for_crsp_firm_annc_m365; by id; run;
proc sort data = for_crsp_vwew_annc_pm5; by id; run;

* Switching back to calling the data files results;
data results5;
    merge master_merged8
        for_crsp_firm_annc_pm5
        for_crsp_firm_annc_m365
        for_crsp_vwew_annc_pm5;
    by id;
run;
```

```
* Check data.;
proc univariate data=master_merged8 freq;
   var filedate_yearq;
run;

proc univariate data=results5 freq;
   var filedate_yearq;
run;

* 12 industry sic dodes;
data results6;
        set results5;
        indus2=int(sic/100);
        indus3=int(sic/10);
*1 NoDur  Consumer NonDurables -- Food, Tobacco, Textiles, Apparel, Leather, Toys;
         if sic ge   0100 and sic le    0999    then ind_code=1;
         if sic ge   2000 and sic le    2399    then ind_code=1;
         if sic ge   2700 and sic le    2749    then ind_code=1;
         if sic ge   2770 and sic le    2799    then ind_code=1;
         if sic ge   3100 and sic le    3199    then ind_code=1;
         if sic ge   3940 and sic le    3989    then ind_code=1;

 *2 Durbl  Consumer Durables -- Cars, TVs, Furniture, Household Appliances;
        if sic ge   2500 and sic le     2519    then ind_code=2;
        if sic ge   2590 and sic le     2599    then ind_code=2;
        if sic ge   3630 and sic le     3659    then ind_code=2;
        if sic ge   3710 and sic le     3711    then ind_code=2;
        if sic ge   3714 and sic le     3714    then ind_code=2;
        if sic ge   3716 and sic le     3716    then ind_code=2;
        if sic ge   3750 and sic le     3751    then ind_code=2;
        if sic ge   3792 and sic le     3792    then ind_code=2;
        if sic ge   3900 and sic le     3939    then ind_code=2;
        if sic ge   3990 and sic le     3999    then ind_code=2;

* 3 Manuf  Manufacturing -- Machinery, Trucks, Planes, Off Furn, Paper, Com Printing;
        if sic ge    2520 and sic le    2589    then ind_code=3;
        if sic ge    2600 and sic le    2699    then ind_code=3;
        if sic ge    2750 and sic le    2769    then ind_code=3;
        if sic ge    3000 and sic le    3099    then ind_code=3;
        if sic ge    3200 and sic le    3569    then ind_code=3;
        if sic ge    3580 and sic le    3629    then ind_code=3;
        if sic ge    3700 and sic le    3709    then ind_code=3;
        if sic ge    3712 and sic le    3713    then ind_code=3;
        if sic ge    3715 and sic le    3715    then ind_code=3;
        if sic ge    3717 and sic le    3749    then ind_code=3;
        if sic ge    3752 and sic le    3791    then ind_code=3;
        if sic ge    3793 and sic le    3799    then ind_code=3;
        if sic ge    3830 and sic le    3839    then ind_code=3;
        if sic ge    3860 and sic le    3899    then ind_code=3;

* 4 Enrgy  Oil, Gas, and Coal Extraction and Products;
        if sic ge     1200 and sic le   1399    then ind_code=4;
        if sic ge     2900 and sic le   2999    then ind_code=4;

* 5 Chems  Chemicals and Allied Products;
        if sic ge     2800 and sic le   2829    then ind_code=5;
        if sic ge     2840 and sic le   2899    then ind_code=5;

* 6 BusEq  Business Equipment -- Computers, Software, and Electronic Equipment;
        if sic ge    3570 and sic le    3579    then ind_code=6;
        if sic ge    3660 and sic le    3692    then ind_code=6;
        if sic ge    3694 and sic le    3699    then ind_code=6;
        if sic ge    3810 and sic le    3829    then ind_code=6;
        if sic ge    7370 and sic le    7379    then ind_code=6;

* 7 Telcm  Telephone and Television Transmission;
```

```
          if sic ge    4800 and sic le   4899    then ind_code=7;

* 8 Utils  Utilities;
          if sic ge    4900 and sic le   4949    then ind_code=8;

* 9 Shops  Wholesale, Retail, and Some Services (Laundries, Repair Shops);
         if sic ge    5000 and sic le    5999    then ind_code=9;
         if sic ge    7200 and sic le    7299    then ind_code=9;
         if sic ge    7600 and sic le    7699    then ind_code=9;

*10 Hlth   Healthcare, Medical Equipment, and Drugs;
          if sic ge   2830 and sic le    2839    then ind_code=10;
          if sic ge   3693 and sic le    3693    then ind_code=10;
          if sic ge   3840 and sic le    3859    then ind_code=10;
          if sic ge   8000 and sic le    8099    then ind_code=10;

*11 Money  Finance;
          if sic ge   6000 and sic le    6999    then ind_code=11;

*12 Other  Other -- Mines, Constr, BldMt, Trans, Hotels, Bus Serv, Entertainment;
          if sic > .     and ind_code = .    then ind_code=12;
run;

proc univariate data=results6 freq;
   var filedate_yearq;
run;

* Merge in institutional investor holdings.;
proc sql;
    create table results7 as
    select a.*,
        b.ii_shares_pct
    from results6 as a
    left join homed.iiholdings as b
    on (a.cusip6 = b.cusip6 and a.filedate_yearq = b.yearq);
quit;

proc univariate data=results7 freq;
   var filedate_yearq;
run;

* Check for duplicates;
proc sort data = results7 nodupkey; by id; run;

* Merge in analyst following data;
libname ibes 'C:\data\shareholder_contracts';
* Create yearq avg;
data ibes2;
    set ibes.eps_consensus_unadjusted_1994;
    if year(date) < 1993 then delete;
    yearq = (year(date)*10)+qtr(date);
    if missing(permno) then delete;
run;

proc univariate data=ibes2 freq;
   var yearq;
run;

proc sql;
    create table ibes3 as
    select permno, yearq, avg(num_forecast) as analyst_following
    from ibes2 as a
    group by permno, yearq;
quit;

data results8;
```

```
    set results7;
    ibes_filedate_yearq = filedate_yearq;
    format ibes_filedate_yearq date9.;
    if missing(ii_shares_pct) = 1 then ii_shares_pct = 0;
run;

proc univariate data=results8 freq;
   var filedate_yearq;
run;

proc univariate data=results8 freq;
   var ibes_filedate_yearq;
run;

* Merge analyst following to main data set;
proc sql;
    create table results9 as
    select a.*,
        b.analyst_following
    from results8 as a
    left join ibes3 as b
    on (a.lpermno = b.permno and a.ibes_filedate_yearq = b.yearq);
quit;
* End merge in analyst following;

proc means; var analyst_following; run;

* Check for duplicates;
proc sort data = results9 nodupkey; by id; run;

* Get distance between investor and target zip codes;
data results10;
    set results9;
    * Reformat both zip codes to make sure you can use them in the distance function below.;
    target_zip2 = compress(target_zip,,"stiap");
    target_zip2 = substr(target_zip2,1,5);
    if length(target_zip2) < 5 then target_zip2 = .;
    target_zip3 = target_zip2;
    filer_zip2 = compress(filer_zip,,"stiap");
    filer_zip2 = substr(filer_zip2,1,5);
    if length(filer_zip2) < 5 then filer_zip2 = .;
    filer_zip3 = filer_zip2;
    * Create variable for distance between filer and target;
    zip_distance=zipcitydistance(substr(target_zip3,1,5),substr(filer_zip3,1,5));
    if substr(target_zip3,1,5) = substr(filer_zip3,1,5) then zip_distance = 0;
    if missing(analyst_following) = 1 then analyst_following = 0;
run;

* Check data.;
proc means data = results10 nway;
    class raw_year;
    var zip_distance;
run;

* Create new variables for later use.;
data results11;
    set results10;
    *  Create indicator for shareholder contract;
    iagreement = (cooperation = 1 | exchange = 1 | investment_agreement = 1 | investment_contract = 1 | investor_agreement = 1 | investor_contract = 1 | rights = 1 | shareholder_agreement = 1 | shareholder_contract = 1 | stockholder_agreement = 1 | stockholder_contract = 1 | support_agreement = 1);
run;

* Preview data;
```

```
proc means; var iagreement; run;

* Pull date when firm first appeared on CRSP to get company age;
libname crspsk "C:\data\shareholder_contracts";
proc sort data = crspsk.stocknames; by permno st_date; run;
data stocknames2;
    set crspsk.stocknames;
    by permno st_date;
    if first.permno ^= 1 then delete;
run;

proc sql;
    create table results15 as
    select distinct a.*, b.st_date
    from results11 as a
    left join stocknames2 as b
    on (a.lpermno = b.permno) and (b.st_date <= a.filedate <= b.end_date);
quit;

data master;
    set results15;
run;

* Make all variable names lowercase;
options mprint;
%macro lowcase(dsn);
     %let dsid=%sysfunc(open(&dsn));
     %let num=%sysfunc(attrn(&dsid,nvars));
     %put &num;
     data &dsn;
           set &dsn(rename=(
        %do i = 1 %to &num;
        %let var&i=%sysfunc(varname(&dsid,&i));      /*function of varname returns the name of a SAS data set variable*/
        &&var&i=%sysfunc(lowcase(&&var&i))         /*rename all variables*/
        %end;));
        %let close=%sysfunc(close(&dsid));
  run;
%mend lowcase;

%lowcase(master)

data homed.master;
    set master;
run;

* Master Stata file;
PROC EXPORT DATA=homed.master
            OUTFILE= "C:\data\shareholder_contracts\master.dta"
            DBMS=STATA REPLACE;
RUN;
```