```
*** file A2a;

libname     dataset     'YOUR_DIRECTORY';



***-----------------------------------------------------------------***
*
*       put togther compustat dataset;
*
***-----------------------------------------------------------------***;


*-------------------------------*;
    %macro getcomp;

        * pull data from compustat;
        proc sql;
            create  table comp as
            select  a.gvkey, a.datadate, a.fyearq, a.rdq, a.fqtr,(a.fyearq * 10) + a.fqtr as fyearqtr, a.atq, a.prccq
            from    comp.fundq a
            where   ((indfmt  = 'INDL') and (consol = 'C') and (popsrc = 'D') and (datafmt = 'STD'))
                and 2008 ge year(datadate) ge 1995          /* set a wide window to start */
            order   by gvkey, fyearqtr, atq descending;
        quit;

        * eliminate small number of dups;
        proc sort data= comp    nodupkey;
            by gvkey fyearqtr;
        run;

        * manually figure out future year-ends and make a couple other variables;
        data comp;
            set comp;
            if(fqtr < 4) then do;
                fyearqtr_ye1 = (fyearq * 10) + 4;
                fyearqtr_ye2 = ((fyearq + 1) * 10) + 4;
                fyearqtr_ye3 = ((fyearq + 2) * 10) + 4;
            end;
            if(fqtr = 4) then do;
                fyearqtr_ye1 = ((fyearq + 1) * 10) + 4;
                fyearqtr_ye2 = ((fyearq + 2) * 10) + 4;
                fyearqtr_ye3 = ((fyearq + 3) * 10) + 4;
            end;

            firmq   = gvkey || fyearqtr;
            rdq_crsp = rdq;
            attrib _all_ label='';

            format rdq_crsp date9.;
        run;

        * this macro gets the rdq trade date;
        %crsp_day(comp, comp, rdq_crsp);


        * get datadates for ye1 ye2 ye3;
        data temp;
            set comp;
        run;

        proc sql;
            create  table comp as
            select  a.*, b.datadate as datadate_ye1,
                    c.datadate as datadate_ye2, d.datadate as datadate_ye3
            from    comp as a left join temp as b
            on      a.gvkey = b.gvkey
                and a.fyearqtr_ye1 = b.fyearqtr

            left join temp as c
            on      a.gvkey = c.gvkey
                and a.fyearqtr_ye2 = c.fyearqtr

            left join temp as d
            on      a.gvkey = d.gvkey
```

```
                and a.fyearqtr_ye3 = d.fyearqtr ;
        quit;


        * keep only relevant years and save to perm file;
        data dataset.pull_analyst2_data_comp;
            set comp;
            if(1997 <= year(rdq) <= 2004);
            if(atq > .);
        run;

    %mend;
*   %getcomp;
*------------------------------*;


*** bring in data;
    data comp;
        set dataset.pull_analyst2_data_comp;
    run;



***-----------------------------------------------------------------***
*
*       get permnos and pull historic cusip;
*
***-----------------------------------------------------------------***;

*------------------------------*;
*** get permnos;
    %macro permno;

        proc sql;
            create  table dataset.pull_analyst2_data_permno as
            select  distinct a.firmq, b.lpermno as permno
            from    comp as a left join crsp.ccmxpf_linktable
                    (where=(LINKTYPE in ("LU","LC","LD","LN","LS","LX")
                    and usedflag=1
                    and linkprim IN ("C", "P"))) as b
            on      a.gvkey = b.gvkey
                and (b.LINKDT <= a.rdq or b.linkdt = .B)
                and (b.LINKENDDT >= a.rdq or b.linkenddt = .E)
            order   by firmq;
        quit;

        proc print data=dataset.pull_analyst2_data_permno   (obs=20);   run;

    %mend permno;
*   %permno;
*------------------------------*;

*** bring back;
    proc sql;
        create  table comp as
        select  a.*, b.permno
        from    comp as a left join dataset.pull_analyst2_data_permno as b
        on      a.firmq = b.firmq
        order   by firmq;
    quit;



*------------------------------*;
*** get historic cusips and exchange codes;
    %macro cusip;

        * pull set of unique permno & rdq combinations, nonmissing only;
        proc sort data= comp (keep= rdq permno where=(permno > . and rdq > .)) nodupkey out=rdq;
            by permno rdq ;
        run;

        proc sql;
            create table cusips as
            select a.*, b.ncusip as cusip, b.shrcd
```

```
            from rdq a, dx.names b
            where   a.permno = b.permno
                and b.date <= a.rdq <= b.nameendt;
        quit;

        * eliminate dups and save to perm set;
        proc sort data= cusips  nodupkey out=dataset.pull_analyst2_data_cusips;
            by permno rdq;
        run;

    %mend cusip;
*   %cusip;
*------------------------------*;


*** merge back;
    proc sql;
        create  table comp as
        select  a.*, b.cusip, b.shrcd
        from    comp as a left join dataset.pull_analyst2_data_cusips as b
        on      a.permno = b.permno
            and a.rdq = b.rdq
        order   by firmq;
    quit;



***-----------------------------------------------------------------***
*
*       pull ibes rdq and time and ue;
*
***-----------------------------------------------------------------***;


*--------------------------*;
*** pull the ibes stuff ;
    %macro getibes;

        * get list of unique cusips and period-ends;
        proc sort data= comp (keep= cusip datadate rdq)     nodupkey    out=cusip (where=(cusip ne ''));
            by cusip datadate;
        run;

        * get ibes rdq and actual eps;
        proc sql;
            create  table actu as
            select  a.*, b.anndats as rdq_ibes, b.value as actual_eps_ibes
            from    cusip a, ibes.actu_epsus b
            where   a.cusip = b.cusip
                and (a.datadate - 7) le b.pends le (a.datadate + 7)
                and b.pdicity = 'QTR'
                and b.measure = 'EPS';
        quit;

        * grab consensus where available;
        proc sql;
            create  table ibes as
            select  a.*, b.numest, b.medest, b.stdev, b.statpers
            from    actu as a left join ibes.statsumu_epsus b
            on      a.cusip = b.cusip
                and (a.datadate - 7) le b.fpedats le (a.datadate + 7)
                and b.fiscalp = 'QTR'
                and b.measure = 'EPS'
                and a.rdq > b.statpers > (a.rdq - 100)      /* only if consensus 100 days or newer */
                and b.statpers > .
            group   by a.cusip, a.datadate
            having  b.statpers = max(b.statpers);
        quit;

        * sort and save;
        proc sort data=ibes     nodupkey    out=dataset.pull_analyst2_data_ibes;
            by cusip datadate;
        run;

        proc print data=ibes    (obs=10);   run;
```

```
    %mend;
*   %getibes;
*-------------------------*;


*** merge back;
    proc sql;
        create  table comp as
        select  a.*, b.rdq_ibes, b.actual_eps_ibes, b.numest as ea_numest, b.medest as ea_medest, b.rdq_ibes - a.rdq as rdq_diff
        from    comp as a left join dataset.pull_analyst2_data_ibes as b
        on      a.cusip = b.cusip
            and a.datadate = b.datadate
        order   by firmq;
    quit;


***-----------------------------------------------------------------***
*
*       cut down to make faster for the rest of this;
*
***-----------------------------------------------------------------***;


*** cut;
    data comp;
        set comp;

        * drop if before 1997;
        if(year(rdq) > 1996);

        if(rdq_ibes = .) then delete;
        rdq_diff_abs = abs(rdq_diff);
        if(rdq_ibes ne rdq) then delete;

        rdq_diff_binary = 0;
        if(rdq_diff ne 0) then rdq_diff_binary = 1;

        attrib _all_ label='';
        drop rdq_diff rdq_diff_binary rdq_diff_abs;
    run;


***-----------------------------------------------------------------***
*
*       get prc and shares right before rdq;
*
***-----------------------------------------------------------------***;

*-------------------------------*;
    %macro prc_rdq;

        %crsp_day ( comp, temp, rdq_crsp, offset = -2 );


        proc sql;
            create  table dataset.pull_analyst2_data_prc_rdq as
            select  a.firmq, abs(b.prc) as prc_rdq, (abs(b.prc) * b.shrout) / 1000 as mve_rdq
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.rdq_crsp = b.date;
        quit;

    %mend prc_rdq;
*   %prc_rdq;
*------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.prc_rdq, b.mve_rdq
        from    comp as a left join dataset.pull_analyst2_data_prc_rdq as b
        on      a.firmq = b.firmq;
    quit;
```

```
***-----------------------------------------------------------------***
*
*       pull ibes actuals for future yearends;
*
***-----------------------------------------------------------------***;


*--------------------------*;
*** pull the ibes stuff ;
    %macro getibes2;

        * get year1;
        proc sql;
            create  table actu as
            select  a.firmq, a.cusip, a.datadate_ye1, a.datadate_ye2, a.datadate_ye3,
                    b.anndats as rdq_ibes_ye1, b.value as actual_eps_ye1
            from    comp as a left join ibes.actu_epsus b
            on  a.cusip = b.cusip
                and (a.datadate_ye1 - 7) le b.pends le (a.datadate_ye1 + 7)
                and b.pdicity = 'ANN'
                and b.measure = 'EPS'
                and a.cusip ne '' and a.datadate_ye1 > .;
        quit;

        proc sort data=actu nodupkey;
            by firmq;
        run;

        * get year2;
        proc sql;
            create  table actu2 as
            select  a.*, b.anndats as rdq_ibes_ye2, b.value as actual_eps_ye2
            from    actu as a left join ibes.actu_epsus as b
            on      a.cusip = b.cusip
                and (a.datadate_ye2 - 7) le b.pends le (a.datadate_ye2 + 7)
                and b.pdicity = 'ANN'
                and b.measure = 'EPS'
                and a.cusip ne '' and a.datadate_ye2 > .;
        quit;

        proc sort data=actu2    nodupkey;
            by firmq;
        run;

        * get year3;
        proc sql;
            create  table actu3 as
            select  a.*, b.anndats as rdq_ibes_ye3, b.value as actual_eps_ye3
            from    actu2 as a left join ibes.actu_epsus as b
            on      a.cusip = b.cusip
                and (a.datadate_ye3 - 7) le b.pends le (a.datadate_ye3 + 7)
                and b.pdicity = 'ANN'
                and b.measure = 'EPS'
                and a.cusip ne '' and a.datadate_ye3 > .;
        quit;

        proc sort data=actu3    nodupkey;
            by firmq;
        run;

        * clean up;
        data dataset.pull_analyst2_data_ibes2;
            set actu3;
            keep firmq rdq_ibes_ye1 rdq_ibes_ye2 rdq_ibes_ye3
                    actual_eps_ye1 actual_eps_ye2 actual_eps_ye3;
            attrib _all_ label='';
        run;

        proc means data=dataset.pull_analyst2_data_ibes2;

    %mend;
*   %getibes2;
```

```
*-------------------------*;


*** merge back;
    proc sql;
        create  table comp as
        select  a.*, b.*
        from    comp as a left join dataset.pull_analyst2_data_ibes2 as b
        on      a.firmq = b.firmq;
    quit;


***-----------------------------------------------------------------***
*
*       identify all analysts who should be covering the firm
*       identify as anyone making at least one annual forecast in year before
*       and in year after the rdq;
*
***-----------------------------------------------------------------***;

*------------------------------*;
    %macro analysts;

    *** make a list of unique cusips in dataset;
        proc sort data=comp nodupkey    out=cusip (keep=cusip);
            by cusip;
        run;


    *** pull all analysts making any annual forecasts;
        proc sql;
            create  table forecast as
            select  a.*, b.analys as analys, b.estimator, b.anndats, b.revdats
            from    cusip as a left join ibes.detu_epsus as b
            on      a.cusip = b.cusip
                and b.fpi in ('1','2','3','4','5')
                and ((1994 <= year(revdats) <= 2007) or (1994 <= year(anndats) <= 2007));
        quit;

        * make analyst_estimator code;
        data all;
            set  forecast ;
            analys2 = estimator||analys;
        run;


    *** identify analysts who should be covering the firm;

        * identify analysts who made or confirmed a forecast in year before rdq;
        proc sql;
            create  table before as
            select  a.firmq, a.rdq, a.cusip, b.analys2, b.anndats as action_date_before,
                    b.estimator, b.analys
            from    comp as a left join all as b
            on      a.cusip = b.cusip
                and (((a.rdq - 366) <= b.anndats < a.rdq)
                 or ((a.rdq - 366) <= b.revdats < a.rdq))
            order   by firmq, analys2, action_date_before descending;
        quit;

        proc sort data=before   nodupkey;
            by firmq analys2;
        run;

        * identify those who have an action in year after;
        proc sql;
            create  table after as
            select  a.*, b.anndats as action_date_after
            from    before as a left join all as b
            on      a.cusip = b.cusip
                and a.analys2 = b.analys2
                and ((a.rdq <= b.anndats <= (a.rdq + 366))
                 or (a.rdq <= b.revdats <= (a.rdq + 366)))
            order   by firmq, analys2, action_date_after;
        quit;
```

```
        proc sort data=after    nodupkey;
            by firmq analys2;
        run;


    *** keep only if in both and save;
        data dataset.pull_analyst2_data_analysts;
            set after;
            if(action_date_before * action_date_after > .);
            attrib _all_ label='';
            drop rdq cusip action_date_before action_date_after;
        run;

        proc print data=dataset.pull_analyst2_data_analysts (obs=10);
        proc means data=dataset.pull_analyst2_data_analysts;

    %mend analysts;
*   %analysts;
*------------------------------*;


*** put back and create uniqueid;
    proc sql;
        create  table comp as
        select  a.*, b.analys2, b.estimator, b.analys
        from    comp as a left join dataset.pull_analyst2_data_analysts as b
        on      a.firmq = b.firmq;
    quit;

    data comp;
        set comp;
        uniqueid = firmq||analys2;
    run;



***-----------------------------------------------------------------***
*
*       get broker and analyst ("bran") names from the IBES Translation File (file bran2);
*
***-----------------------------------------------------------------***;

*------------------------------*;
    %macro bran;

    *** load broker data;
        data bran;
            set dataset.bran2;
            if(usfirm = 1 and baname ne '');
            drop usfirm;
        run;

        proc sort data=bran nodupkey;
            by bacode baindi;;
        run;    * 0.0035% lost.  usually close name spelling differences. ignore;


    *** get broker name;
        proc sql;
            create  table bran2 as
            select  a.uniqueid, a.analys, a.estimator, b.baname as broker_name
            from    comp as a left join bran as b
            on      a.estimator = b.bacode
                and b.baindi = 'B';
        quit;


    *** get analyst name;
        proc sql;
            create  table bran3 as
            select  a.*, b.baname as analyst_name
            from    bran2 as a left join bran as b
            on      a.analys = b.bacode
                and b.baindi = 'A';
```

```
        quit;

        proc sort data=bran3    nodupkey    out=dataset.pull_analyst2_data_bran (keep= uniqueid analyst_name broker_name);
            by uniqueid;
        run;    * checked and none lost;

        proc print data=dataset.pull_analyst2_data_bran (obs=10);

    %mend bran;
*   %bran;
*-------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*,b.analyst_name, b.broker_name
        from    comp as a left join dataset.pull_analyst2_data_bran as b
        on      a.uniqueid = b.uniqueid;
    quit;


*** drop if dont have both broker and analyst name;
    data comp;
        set comp;
        if(analyst_name = '' or broker_name = '') then delete;
        drop broker_name analyst_name;
    run;


***-----------------------------------------------------------------***
*
*       drop low coverage and create final dataset for data collection
*
***-----------------------------------------------------------------***;


*** drop if analyst doesn't follow at least 2 firms in 4 quarters;
    proc sql;
        create  table counts as
        select  distinct a.analys, count(distinct a.gvkey) as firm_cnt, count(a.firmq) as fqtrs_cnt
        from    comp as a
        group   by analys;
    quit;

    proc sql;
        create  table comp as
        select  a.*, b.firm_cnt, b.fqtrs_cnt
        from    comp as a left join counts as b
        on      a.analys = b.analys;
    quit;

    data comp;
        set comp;
        if(firm_cnt < 2 or fqtrs_cnt < 4) then delete;
    run;


*** drop if fewer than three analysts in a firm-quarter;
    proc sql;
        create  table counts as
        select  distinct a.firmq, count(distinct a.analys) as analyst_cnt
        from    comp as a
        group   by firmq;
    quit;

    proc sql;
        create  table comp as
        select  a.*, b.analyst_cnt
        from    comp as a left join counts as b
        on      a.firmq = b.firmq;
    quit;

    data comp;
        set comp;
        if(analyst_cnt < 2) then delete;
```

```
    run;



*** save set for data collection;
*   data dataset.pull_analyst2_analysts_collect;
*       set comp;
*       keep estimator analys rdq;
*   run;


*** SAVE FINAL DATASET TO GO INTO PARTB;
    proc means data=comp;

    data dataset.pull_analyst2_data_final;
        set comp;
    run;

    endsas;
```