```
*** file A2c;

*** setup libnames and gvkeys for beta testing;
    libname     dataset     'YOUR_DIRECTORY';
    libname     this        'YOUR_DIRECTORY';


*** set a macro variable for the default measurement window after the earnings announcement;
*** NOTE:   an early version of the paper measured analyst activity over a 7-day window, which is why
***         this is set to 7 days.  Code further below changes the activity variables to 3-day windows;

    %let        _eawindow   = 7;



***-----------------------------------------------------------------***
*
*       load dataset and then bring in bring in hand-collected data
*
***-----------------------------------------------------------------***;

*** load data from part1;
    data comp;
        set dataset.pull_analyst2_data_final;
    run;


*** get hand-collected location data;
    proc sql;
        create  table comp as
        select  a.*, b.zip as zip,
                b.latitude as alat, b.longitude as along
        from    comp as a left join dataset.qea_locations as b
        on      a.estimator = b.estimator
            and a.analys = b.analys
            and a.rdq = b.rdq;
    quit;


*** drop if missing location;
    data comp;
        set comp;
        if(zip = '') then delete;
    run;


*** setup 3-day measurement window;
    data comp;
        set comp;
        win2_end = rdq_crsp;
        format win2_end date9.;
    run;

    %crsp_day(comp, comp, win2_end, offset = 2);


*** output zip code list to use in cutting down weather dataset;
*   proc sort data=comp nodupkey    out=this.zip_codes (keep=zip );
*       by zip;
*   run;




***-----------------------------------------------------------------***
*       calculate DELAY and MAKE_FCAST
*       get delay on first ye forecast for future ye;
*       any fcast is fine.  does not have to be diluted, etc.;
*
***-----------------------------------------------------------------***;

*-------------------------------*;
    %macro delay;

        proc sql;
            create  table fcast as
            select  a.uniqueid, a.cusip, a.rdq, a.analys, a.estimator, b.anndats as first_fcast_date
            from    comp as a left join ibes.detu_epsus as b
            on      a.cusip = b.cusip
                and a.estimator = b.estimator
                and a.analys = b.analys

                /* must be within window after rdq */
                and a.rdq <= b.anndats <= (a.rdq + &_eawindow)

                /* must be an annual measure */
                and b.fpi in ('1','2','3','4','5')

                /* only keep if one of first upcoming year-ends */
                and (((a.datadate_ye1 - 7) <= b.fpedats <= (a.datadate_ye1 + 7))
                or   ((a.datadate_ye2 - 7) <= b.fpedats <= (a.datadate_ye2 + 7))
                or   ((a.datadate_ye3 - 7) <= b.fpedats <= (a.datadate_ye3 + 7)))

            order   by uniqueid, first_fcast_date;
        quit;

```

```
        proc sort data=fcast    nodupkey;
            by uniqueid;
        run;


        * clean up and save;
        data dataset.pull_analyst_pb_data_delay;
            set fcast;
            make_fcast = 0;
            if(first_fcast_date > .) then make_fcast = 1;
            delay = first_fcast_date - rdq;
            drop cusip rdq analys estimator;
        run;

    %mend delay;
*   %delay;
*-------------------------------*;


*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.first_fcast_date, b.make_fcast
        from    comp as a left join dataset.pull_analyst_pb_data_delay as b
        on      a.uniqueid = b.uniqueid;
    quit;


***-----------------------------------------------------------------***
*
*       get values of first forecast after EA for each period-end
*
***-----------------------------------------------------------------***;

*-------------------------------*;
    %macro fcast_val;

        * get for ye1;
        proc sql;
            create  table fcast as
            select  a.uniqueid, a.cusip, a.rdq, a.analys, a.estimator, a.datadate_ye1, a.datadate_ye2,
                    a.datadate_ye3, b.value as fcast_value_ye1, b.anndats as fcast_date_ye1
            from    comp as a left join ibes.detu_epsus as b
            on      a.cusip = b.cusip
                and a.estimator = b.estimator
                and a.analys = b.analys

                /* must be within window after rdq */
                and a.rdq <= b.anndats <= (a.rdq + &_eawindow)

                /* must be an annual measure, EPS, diluted */
                and b.fpi in ('1','2','3','4','5')
                and b.measure = 'EPS'
                and b.report_curr = 'USD'
                and b.pdf = 'D'

                /* only keep if ye1 */
                and ((a.datadate_ye1 - 7) <= b.fpedats <= (a.datadate_ye1 + 7))

            order   by uniqueid, fcast_date_ye1;
        quit;

        proc sort data=fcast    nodupkey;
            by uniqueid;
        run;

        * get for ye2;
        proc sql;
            create  table fcast2  as
            select  a.*, b.value as fcast_value_ye2, b.anndats as fcast_date_ye2
            from    fcast as a left join ibes.detu_epsus as b
            on      a.cusip = b.cusip
                and a.estimator = b.estimator
                and a.analys = b.analys

                /* must be within window after rdq */
                and a.rdq <= b.anndats <= (a.rdq + &_eawindow)

                /* must be an annual measure, EPS, diluted */
                and b.fpi in ('1','2','3','4','5')
                and b.measure = 'EPS'
                and b.report_curr = 'USD'
                and b.pdf = 'D'

                /* only keep if ye2 */
                and ((a.datadate_ye2 - 7) <= b.fpedats <= (a.datadate_ye2 + 7))

            order   by uniqueid, fcast_date_ye2;
        quit;

        proc sort data=fcast2   nodupkey    ;
            by uniqueid;
        run;

        * get for ye3;
        proc sql;
```

```
            create  table fcast3 (drop= cusip rdq analys estimator datadate_ye1 datadate_ye2 datadate_ye3) as
            select  a.*, b.value as fcast_value_ye3, b.anndats as fcast_date_ye3
            from    fcast2 as a left join ibes.detu_epsus as b
            on      a.cusip = b.cusip
                and a.estimator = b.estimator
                and a.analys = b.analys

                /* must be within window after rdq */
                and a.rdq <= b.anndats <= (a.rdq + &_eawindow)

                /* must be an annual measure, EPS, diluted */
                and b.fpi in ('1','2','3','4','5')
                and b.measure = 'EPS'
                and b.report_curr = 'USD'
                and b.pdf = 'D'

                /* only keep if ye3 */
                and ((a.datadate_ye3 - 7) <= b.fpedats <= (a.datadate_ye3 + 7))

            order   by uniqueid, fcast_date_ye3;
        quit;

        proc sort data=fcast3   nodupkey    out=dataset.pull_analyst_pb_data_fcast_val;
            by uniqueid;
        run;


    %mend fcast_val;
*   %fcast_val;
*------------------------------*;

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



***-----------------------------------------------------------------***
*
*       calculate fcast_error and fcast_bias;
*
***-----------------------------------------------------------------***;

*------------------------------*;
    %macro fe;

        * save temp set;
        data temp;
            set comp;
        run;

        %crsp_day ( temp, temp, fcast_date_ye1, previous = y );
        %crsp_day ( temp, temp, rdq_ibes_ye1, previous = y );
        %crsp_day ( temp, temp, fcast_date_ye2, previous = y );
        %crsp_day ( temp, temp, rdq_ibes_ye2, previous = y );
        %crsp_day ( temp, temp, fcast_date_ye3, previous = y );
        %crsp_day ( temp, temp, rdq_ibes_ye3, previous = y );
        %crsp_day ( temp, temp, rdq_crsp, offset = -2 );

        * get split factors on fcast dates;
        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as fcast_date_ye1_cfacpr, b.cfacshr as fcast_date_ye1_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.fcast_date_ye1= b.date;
        quit;

        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as fcast_date_ye2_cfacpr, b.cfacshr as fcast_date_ye2_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.fcast_date_ye2= b.date;
        quit;

        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as fcast_date_ye3_cfacpr, b.cfacshr as fcast_date_ye3_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.fcast_date_ye3= b.date;
        quit;

        * get split factors on ye1 ye2 rdq;
        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as rdq_ibes_ye1_cfacpr, b.cfacshr as rdq_ibes_ye1_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.rdq_ibes_ye1= b.date;
        quit;
```

```

        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as rdq_ibes_ye2_cfacpr, b.cfacshr as rdq_ibes_ye2_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.rdq_ibes_ye2= b.date;
        quit;

        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as rdq_ibes_ye3_cfacpr, b.cfacshr as rdq_ibes_ye3_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.rdq_ibes_ye3= b.date;
        quit;

        * get split factor on initial rdq;
        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as rdq_cfacpr, b.cfacshr as rdq_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.rdq_crsp= b.date;
        quit;

        * calculate stuff;
        data dataset.pull_analyst_pb_data_fe;
            set temp;

            fcast_value_ye1_adj = fcast_value_ye1 / fcast_date_ye1_cfacshr;
            fcast_value_ye2_adj = fcast_value_ye2 / fcast_date_ye2_cfacshr;
            fcast_value_ye3_adj = fcast_value_ye3 / fcast_date_ye3_cfacshr;

            actual_eps_ye1_adj  = actual_eps_ye1 / rdq_ibes_ye1_cfacshr;
            actual_eps_ye2_adj  = actual_eps_ye2 / rdq_ibes_ye2_cfacshr;
            actual_eps_ye3_adj  = actual_eps_ye3 / rdq_ibes_ye3_cfacshr;

            fe_ye1_raw = (fcast_value_ye1_adj - actual_eps_ye1_adj) * rdq_cfacshr;
            fe_ye2_raw = (fcast_value_ye2_adj - actual_eps_ye2_adj) * rdq_cfacshr;
            fe_ye3_raw = (fcast_value_ye3_adj - actual_eps_ye3_adj) * rdq_cfacshr;

            fe_ye1 = fe_ye1_raw / prc_rdq;
            fe_ye2 = fe_ye2_raw / prc_rdq;
            fe_ye3 = fe_ye3_raw / prc_rdq;

            keep uniqueid fe_ye1_raw fe_ye2_raw fe_ye3_raw fe_ye1 fe_ye2 fe_ye3;
        run;

    %mend fe;
*   %fe;
*------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.fe_ye2_raw, b.fe_ye2
        from    comp as a left join dataset.pull_analyst_pb_data_fe as b
        on      a.uniqueid = b.uniqueid
            and a.make_fcast = 1;               /* fe made all obs but only want if in fcast window */;
    quit;


***-----------------------------------------------------------------***
*
*       make converge by doing EPS forecast one day prior to each rec;
*
***-----------------------------------------------------------------***;

*------------------------------*;
    %macro converge;

    *** make separate datasets of ye1 and ye2, then stack;
        data ye1;
            set comp;
            fcast_date = fcast_date_ye1;
            datadate_ye = datadate_ye1;
            if(datadate_ye * fcast_date > .);
            keep cusip fcast_date datadate_ye;
        run;

        data ye2;
            set comp;
            fcast_date = fcast_date_ye2;
            datadate_ye = datadate_ye2;
            if(datadate_ye * fcast_date > .);
            keep cusip fcast_date datadate_ye;
        run;

        data ye3;
            set ye1 ye2;
            id3 = cusip||fcast_date||datadate_ye;
            format fcast_date datadate_ye date9.;
        run;

        proc sort data=ye3  nodupkey;
```

```
            by id3;
        run;


    *** get forecasts for those period-ends made or confirmed within 180 days of period-end;
        proc sql;
            create  table fcast as
            select  a.*, b.analys, b.estimator, b.anndats, b.revdats, b.value
            from    ye3 as a left join ibes.detu_epsus as b
            on      a.cusip = b.cusip

                /* only keep if for the correct period */
                and ((a.datadate_ye - 7) <= b.fpedats <= (a.datadate_ye + 7))

                /* must be within the last 180 days before the fcast_date */
                and (a.fcast_date - 180) <= b.anndats < a.fcast_date

                /* must be an annual measure, EPS, diluted */
                and b.fpi in ('1','2','3','4','5')
                and b.measure = 'EPS'
                and b.report_curr = 'USD'
                and b.pdf = 'D'

            order   by id3, estimator, analys, revdats descending, anndats descending;
        quit;

        * just keep last;
        proc sort data=fcast    nodupkey;
            by id3 estimator analys;
        run;


    *** calculate consensus;
        proc means data=fcast   noprint;
            by cusip fcast_date datadate_ye;
            var value;
            output out=dataset.pull_analyst_pb_data_converge    (drop= _type_ _freq_)
                n(value) = recent_cons_n median(value) = recent_cons_val;
        quit;

    %mend converge;
*   %converge;
*-------------------------------*;

*** put back ye2 (do not need ye1);
    proc sql;
        create  table comp as
        select  a.*, b.recent_cons_val as recent_cons_ye2
        from    comp as a left join dataset.pull_analyst_pb_data_converge as b
        on      a.cusip = b.cusip
            and a.datadate_ye2 = b.datadate_ye
            and a.fcast_date_ye2 = b.fcast_date;
    quit;


***-----------------------------------------------------------------***
*
*       get make_recommend and recommend;
*
***-----------------------------------------------------------------***;

*-------------------------------*;
    %macro makerec;

        * make numeric ireccd;
        data recommend;
            set dataset.recddet;
            ireccd_num = ireccd * 1;
            ireccd_num = (-1 * ireccd_num) + 5;
            if(ireccd in ('1','2','3','4','5'));
        run;


        * check had at least one rec before. keep last;
        proc sql;
            create  table rec as
            select  a.uniqueid, a.analys, a.estimator, a.rdq, a.cusip,
                    b.anndats as rec_date_before, b.ireccd_num as ireccd_before, b.revdats as rec_revdate_before
            from    comp as a left join recommend as b
            on      a.cusip = b.cusip
        /*      and a.estimator = b.EMASKCD */              /* mapping of estimator to EMASKCD does not work*/
                and a.analys = b.amaskcd

                /* eliminate tiny fraction of restricted */
                and b.ireccd in ('1','2','3','4','5')

                /* either the anndats or revdats before */
                and ( ((a.rdq - 366) <= b.anndats < a.rdq)
                or    ((a.rdq - 366) <= b.revdats < a.rdq) )

            order   by uniqueid, rec_date_before descending, rec_revdate_before descending;
        quit;

        proc sort data=rec  nodupkey;
            by uniqueid;
        run;
```

```


        * get rec after if within N days;
        proc sql;
            create  table rec2 as
            select  a.*, b.anndats as rec_date_after, b.ireccd_num as ireccd_after, b.revdats as rec_revdate_after
            from    rec as a left join recommend as b
            on      a.cusip = b.cusip
                and a.analys = b.amaskcd

                /* eliminate tiny fraction of restricted */
                and b.ireccd in ('1','2','3','4','5')

                /* either the anndats or revdats before */
                and (a.rdq <= b.anndats <= (a.rdq + &_eawindow))

            order   by uniqueid, rec_date_after, rec_revdate_after;
        quit;

        proc sort data=rec2 nodupkey;
            by uniqueid;
        run;


        * if more than N days after rdq, then delete ireccd_after;
        data dataset.pull_analyst_pb_data_makerec;
            set rec2;

            * if dont have rec before then delete after;
            if(rec_date_before = .) then ireccd_after = .;

            * put in indicator if takes action;
            if(rec_date_before > . )                        then make_recommend = 0;
            if(rec_date_before > . and ireccd_after > .)    then make_recommend = 1;

            * if rec_before is there and rec_after is not, assume confirmed;
            if(ireccd_after = . and make_recommend > .) then ireccd_after = ireccd_before;

            if(make_recommend = .) then delete;

            drop analys estimator rdq cusip fcast_within_year_after rec_date_before rec_revdate_before rec_revdate_after;
        run;

    %mend makerec;
*   %makerec;
*-------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.ireccd_before, b.make_recommend, b.ireccd_after, b.rec_date_after
        from    comp as a left join dataset.pull_analyst_pb_data_makerec as b
        on      a.uniqueid = b.uniqueid;
    quit;


***-----------------------------------------------------------------***
*
*       get make_tprice and tprice;
*
***-----------------------------------------------------------------***;


*-------------------------------*;
    %macro maketprice;

        * get last tprice before;
        proc sql;
            create  table rec as
            select  a.uniqueid, a.analys, a.estimator, a.rdq, a.cusip,
                    b.anndats as tprice_date_before, b.value as tprice_before
            from    comp as a left join dataset.tprice as b
            on      a.cusip = b.cusip
                and a.analys = b.amaskcd

                /* eliminate foreign currency and zero dollar */
                and b.curr = 'USD'
                and b.value > 0

                /* within a year */
                and ((a.rdq - 366) <= b.anndats < a.rdq)

            order   by uniqueid, tprice_date_before descending;
        quit;

        proc sort data=rec  nodupkey;
            by uniqueid;
        run;


        * get tprice after if within N days;
        proc sql;
            create  table rec2 as
            select  a.*, b.anndats as  tprice_date_after, b.value as tprice_after
            from    rec as a left join dataset.tprice as b
            on      a.cusip = b.cusip
```

```
                and a.analys = b.amaskcd

                /* eliminate foreign currency*/
                and b.curr = 'USD'

                /* get first */
                and (a.rdq <= b.anndats <= (a.rdq + &_eawindow))

            order   by uniqueid, tprice_date_after;
        quit;

        proc sort data=rec2 nodupkey;
            by uniqueid;
        run;


        * if more than N days after rdq, then delete ireccd_after;
        data dataset.pull_analyst_pb_data_mtprice;
            set rec2;

            * if dont have a tprice before then delete one after;
            if(tprice_date_before = .) then tprice_after = .;

            * put indicator if takes action;
            if(tprice_date_before > .)                          then make_tprice = 0;
            if(tprice_date_before > . and tprice_after > .)     then make_tprice = 1;

            * if tprice_before is there and tprice_after is not, assume same;
            if(tprice_after = . and make_tprice > .) then tprice_after = tprice_before;

            if(make_tprice = .) then delete;
            drop analys estimator rdq cusip tprice_date_before;
        run;

    %mend maketprice;
*   %maketprice;
*------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.tprice_before, b.make_tprice, b.tprice_after, b.tprice_date_after
        from    comp as a left join dataset.pull_analyst_pb_data_mtprice as b
        on      a.uniqueid = b.uniqueid;
    quit;


***-----------------------------------------------------------------***
*
*       calculate  tprice_error tprice_bias;
*           *** IBES TARGET PRICE IS ALREADY ADJUSTED;
*           *** want to scale by price when made tprice forecast
*
***-----------------------------------------------------------------***;

*------------------------------*;
    %macro tprice_error;

        * save temp set;
        data temp;
            set comp;
            tprice_date_future = tprice_date_after + 365;
            keep uniqueid tprice_date_after tprice_date_future rdq_crsp permno tprice_after prc_rdq tprice_before;
            format tprice_date_future date9.;
        run;

        %crsp_day ( temp, temp, tprice_date_after, previous = y );
        %crsp_day ( temp, temp, tprice_date_future, previous = y );
        %crsp_day ( temp, temp, rdq_crsp, offset = -2 );


        * get split factors on tprice_date_after;
        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as tprice_date_after_cfacpr, b.cfacshr as tprice_date_after_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.tprice_date_after= b.date;
        quit;

        * get split factors on tprice_date_future;
        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as tprice_date_future_cfacpr, b.cfacshr as tprice_date_future_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.tprice_date_future= b.date;
        quit;

        * get split factors on rdq date;
        proc sql;
            create  table temp as
            select  a.*, b.cfacpr as rdq_crsp_cfacpr, b.cfacshr as rdq_crsp_cfacshr
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.rdq_crsp= b.date;
```

```
        quit;

        * get price on tprice_date_future;
        proc sql;
            create  table temp as
            select  a.*, b.prc as tprice_future_actual
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.tprice_date_future = b.date;
        quit;

        * get price on tprice_date_after;
        proc sql;
            create  table temp as
            select  a.*, b.prc as tprice_date_actual
            from    temp as a left join crspq.dsf as b
            on      a.permno = b.permno
                and a.tprice_date_after = b.date;
        quit;

        * calculate stuff;
        data dataset.pull_analyst_pb_data_tp_error;
            set temp;

            tprice_future_actual_adj    = tprice_future_actual / tprice_date_future_cfacpr;
            tprice_error_raw    = (tprice_after - tprice_future_actual_adj) * tprice_date_after_cfacpr;
            tprice_error        = tprice_error_raw / tprice_date_actual;
            tprice_chg          = tprice_after - tprice_before;
            tprice_chg_scale    = (tprice_chg * tprice_date_after_cfacpr) / tprice_date_actual;
            tprice_before_scale = (tprice_before * rdq_crsp_cfacpr) / prc_rdq;
            tprice_after_scale  = (tprice_after * tprice_date_after_cfacpr) / tprice_date_actual;
            keep uniqueid tprice_error_raw tprice_error tprice_chg tprice_chg_scale tprice_before_scale tprice_after_scale;
        run;

    %mend tprice_error;
*   %tprice_error;
*------------------------------*;

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


***-----------------------------------------------------------------***
*
*       get weather;
*
***-----------------------------------------------------------------***;

*------------------------------*;
*** calculate over 3-day windows;
    %macro cloud3d;

    *** set dates;
        data temp;
            set comp;
            weatherdate = win2_end;
            if(. < first_fcast_date     < win2_end)                                       then weatherdate = first_fcast_date;
            if(. < tprice_date_after    < win2_end) and (tprice_date_after  < weatherdate) then weatherdate = tprice_date_after;
            if(. < rec_date_after       < win2_end) and (rec_date_after     < weatherdate) then weatherdate = rec_date_after;
        run;


    *** pull weather and do for rdq start date;
        proc sql;
            create  table weather as
            select  a.uniqueid, b.recdate,
                    b.cloud_mean,   b.rain_mean,    b.wind_mean
            from    temp as a left join dataset.weather_final_anal_20160818 as b
            on      a.zip = b.zip
                and a.rdq <= b.recdate <= a.weatherdate
            order   by uniqueid;
        quit;

        proc means data=weather noprint;
            by uniqueid;
            var cloud_mean;
            output out=dataset.pull_analyst_pb4_data_cloud3d
                mean(cloud_mean rain_mean wind_mean )
                =    cloud_mean rain_mean wind_mean ;
        quit;


    *** get extreme storms locations;
        proc sql;
            create  table temp as
            select  a.uniqueid, a.zip, a.rdq, a.weatherdate, b.state_county_id
            from    temp as a left join dataset.zips_final as b
            on      a.zip = b.zip;
        quit;

    *** take a look at what types of storms happen in event window;
```

```
        proc sql;
            create  table temp2 as
            select  a.*, b.event_type
            from    temp as a , dataset.storm_events_final as b
            having  a.state_county_id = b.state_county_id
                and a.rdq <= b.date <= a.weatherdate
            order   by uniqueid;
        quit;

        data temp2;
            set temp2;

            * delete strictly marine events;
            if(event_type in ('Rip Current' 'Waterspout'))      then    delete;

            * consolidate others;
            if(event_type in ('Flash Flood' 'Flood'))           then    flood2 = 1;
                else                                                    flood2 = 0;
            if(event_type in ('Thunderstorm Wind' 'Lightning')) then    storm2 = 1;
                else                                                    storm2 = 0;
            if(event_type in ('Funnel Cloud' 'Tornado'))        then    tornado2 = 1;
                else                                                    tornado2 = 0;
            if(event_type in ('Hail'))                          then    hail2 = 1;
                else                                                    hail2 = 0;
            if(event_type in ('Heavy Rain'))                    then    rain2 = 1;
                else                                                    rain2 = 0;
            if(flood2 = storm2 = tornado2 = hail2 = rain2 = 0)  then    other = 1;
                else                                                    other = 0;
            xtrm_any = flood2 + storm2 + tornado2 + hail2 + rain2 + other;
            if(xtrm_any > 1) then xtrm_any = 1;
        run;


    *** calculate max over window;
        proc means data=temp2   noprint;
            by uniqueid;
            var flood2;
            output out=dataset.pull_analyst_pb4_data_extreme3d      (drop= _type_ _freq_)
                                    max(flood2 storm2 tornado2 hail2 rain2 other xtrm_any)
                                    = xtrm_flood3d xtrm_storm3d xtrm_torn3d xtrm_hail3d xtrm_rain3d xtrm_other3d xtrm_any3d;
        run;

    %mend cloud3d;
*   %cloud3d;
*------------------------------*;

*** put back weather data;
    proc sql;
        create  table comp as
        select  a.*, b.cloud_mean as cloud3d, b.rain_mean as rain3d, b.wind_mean as wind3d

        from    comp as a left join dataset.pull_analyst_pb4_data_cloud3d as b
        on      a.uniqueid = b.uniqueid;
    quit;

    proc sql;
        create  table comp as
        select  a.*, b.xtrm_any3d
        from    comp as a left join dataset.pull_analyst_pb4_data_extreme3d as b
        on      a.uniqueid = b.uniqueid;
    quit;


***-----------------------------------------------------------------***
*
*       calculate other variables;
*
***-----------------------------------------------------------------***;

*** calc;
    data comp;
        set comp;

        if(make_recommend > .) then  ireccd_chg = ireccd_after - ireccd_before;

        * rescale the forecast error;
        fe_ye2 = fe_ye2 * 100;

        * calculate deviation from consensus;
        fcast_ye2_cons_diff = ((fcast_value_ye2 - recent_cons_ye2) / prc_rdq) * 100;

        * missing xtrm_any3d indicates no extreme weather, so set to zero;
        if(xtrm_any3d = .)                                      then xtrm_any3d = 0;

        attrib _all_ label='';

    run;;




***-----------------------------------------------------------------***
*
*       An early version of the paper measured analyst output over 7 days.
*       In this section, convert the 7-day measurements to 3-day measurements;
*
```

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

*** set the make variables to 0 if the make happens after win2_end;
    data comp;
        set comp;

        * set them up;
        make_recommend_3d       = make_recommend;
        make_tprice_3d          = make_tprice;
        make_fcast_3d           = make_fcast;

        ireccd_chg_3d           = ireccd_chg;
        fe_ye2_3d               = fe_ye2;
        fcast_ye2_cons_diff_3d  = fcast_ye2_cons_diff;
        tprice_error_3d         = tprice_error;


        * clear out ones not in 3d window;
        if(make_recommend_3d = 1) and (rec_date_after > win2_end) then do;
            make_recommend_3d = 0;
        end;

        if(make_tprice_3d = 1) and (tprice_date_after > win2_end) then do;
            make_tprice_3d = 0;
            tprice_error_3d = .;
        end;

        if(make_fcast_3d = 1) and (first_fcast_date > win2_end) then do;
            make_fcast_3d = 0;
        end;

        if(fe_ye2_3d > .) and (fcast_date_ye2 > win2_end)               then fe_ye2_3d = .;
        if(fcast_ye2_cons_diff_3d > .) and (fcast_date_ye2 > win2_end)  then fcast_ye2_cons_diff_3d = .;

        * make a 3-day delay variable;
        if(make_fcast_3d = 1 and first_fcast_date <= win2_end) then delay_3d = intck('WEEKDAY', rdq, first_fcast_date);


        * drop the old 7-day versions;
        drop make_recommend make_tprice make_fcast ireccd_chg fe_ye2 fcast_ye2_cons_diff tprice_error delay;
    run;


***-------------------------------------------------------------***
*
*       cut down, make post-cut vars, winsorize;
*
***-------------------------------------------------------------***;


*** cut down;;
    data comp;
        set comp;

        * must have weather to construct the weather factor;
        if(cloud3d > . & rain3d > . & wind3d > .);

        * must have data on at least one activity measure to be useable;
        if(make_recommend_3d = make_tprice_3d = make_fcast_3d = delay_3d = .) then delete;

    run;


*** if more than one datadate on rdq, keep the latter;
    proc sort data=comp;
        by gvkey analys2 rdq descending datadate;
    run;

    proc sort data=comp     nodupkey;
        by gvkey analys2 rdq ;
    run;


*** create the weather variables;
    data comp;
        set comp;

        * log the weather variables;
        %macro quick1 (var);
            &var._log       = log(&var          + 1);
        %mend quick1 (var);

        %quick1(cloud3d);
        %quick1(wind3d);
        %quick1(rain3d);


        * standardize the weather variables;
        %macro quick1 (var);
            &var._log_std       = &var._log;
        %mend quick1 (var);

        %quick1(cloud3d);
        %quick1(wind3d);
        %quick1(rain3d);
```

```
        proc standard data=comp     mean=0 std=1        out=comp;
            var rain3d_log_std wind3d_log_std cloud3d_log_std ;
        run;


        * create weather factor;
        proc factor noprint data=comp method=principal rotate=v n=1  out=comp (rename=(factor1 = wf3d) );
            var cloud3d_log wind3d_log rain3d_log;
        run;


*** drop variables that are not needed in the Stata dataset;
    data comp;
        set comp;
        drop fcast_date_ye:
            fyearqtr_ye1 fyearqtr_ye2 rdq_crsp rdq_ibes
            rdq_ibes_ye1 rdq_ibes_ye2 actual_eps_ye1 actual_eps_ye2 analys2 datadate_ye1 datadate_ye2
            firm_cnt fqtrs_cnt  actual_eps_ibes ea_medest prccq
            recent_cons_ye2 ea_numest tprice_before tprice_after
            actual_eps_ibes_ye3 rdq_ibes_ye3  fyearqtr_ye3 datadate_ye3
            fcast_value_ye:
            actual_eps_ye3
            analyst_cnt shrcd permno atq alat along


            fe_ye2_raw
            ireccd_before
            ireccd_after
            rec_date_after
            tprice_date_after

            zip
            win2_end
            first_fcast_date
            cloud3d_log     wind3d_log      rain3d_log
            ;

    run;


*** winsor;
    %winsor(dsetin=comp, dsetout=comp, byvar=none,
            vars=

            /* dependent variables */
            delay_3d    fe_ye2_3d   fcast_ye2_cons_diff_3d  tprice_error_3d ireccd_chg_3d

            /* control and descriptive variables */
            mve_rdq   prc_rdq


            /* weather variables */
            cloud3d             wind3d              rain3d
            cloud3d_log_std     wind3d_log_std      rain3d_log_std
            wf3d

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


***-----------------------------------------------------------------***
*
*       save final sets;
*
***-----------------------------------------------------------------***;

*** save;
    data this.Analyst_Data_Final;
        set comp;
    run;

    proc export data=comp   outfile= "Analyst_Data_Final.dta" replace;  run;


*** save a dataset of just observation identifiers for JAR;
    data this.A4_Observations;
        set comp;
        keep gvkey rdq fyearqtr estimator analys;
    run;

endsas;
```