```
*** file M2a;

libname     dataset     'YOUR_DIRECTORY';
libname     this        'YOUR_DIRECTORY';


***-----------------------------------------------------------------***
*
*       get compustat rdq;
*
***-----------------------------------------------------------------***;

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

        * pull data;
        proc sql;
            create  table comp as
            select  a.gvkey, a.datadate, a.fyearq, a.rdq as eadate,  a.fqtr,(a.fyearq * 10) + a.fqtr as fyearqtr,
                    a.atq, a.prccq, a.cshoq, a.seqq, year(a.rdq) as year, a.ibq,
                    a.rdq as eadate_crsp, a.rdq as win2_end, a.rdq as win14_end, a.rdq as win70_end
            from    comp.fundq a
            where   ((indfmt  = 'INDL') and (consol = 'C') and (popsrc = 'D') and (datafmt = 'STD'))
                and a.atq * a.prccq > .
                and 1990 <= year(a.rdq) <= 2014;
        quit;

        proc sort data=comp nodupkey;
            by gvkey fyearqtr;
        run;

        * find the trading day on or after the eadate;
        %crsp_day(comp, comp, eadate_crsp);

        * setup the end of the erc window;
        %crsp_day(comp, comp, win2_end, offset = 2);


        * drop if not in the USA;
        proc sql;
            create  table comp2 as
            select  a.*, b.state, b.loc, substr(b.addzip, 1, 5) as comp_zip
            from    comp as a left join comp.company as b
            on      a.gvkey = b.gvkey;
        quit;

        data comp2;
            set comp2;
            if(loc = 'USA');
            drop state loc;
            firmq       = gvkey||fyearqtr;
            attrib _all_ label='';
        run;


        * eliminate dups and save to perm file;
        proc sort data= comp2   nodupkey    out=dataset.erc3_data_compustat;
            by firmq;
        run;

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

*** bring in data;
    data comp;
        set dataset.erc3_data_compustat;
        mergeall = 1;
        * make a begin date for ERC and also a 75-day pead window ending date;
        win_beg = eadate_crsp;
        win75_end = eadate_crsp;
        format win_beg win75_end date9.;
    run;

    %crsp_day(comp, comp, win75_end, offset = 75);



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

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

        proc sql;
            create  table dataset.erc3_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.eadate or b.linkdt = .B)
                and (b.LINKENDDT >= a.eadate or b.linkenddt = .E)
            order   by firmq;
        quit;

        proc print data=dataset.erc3_data_permno    (obs=20);   run;

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

*** bring back;
    data comp;
        merge comp (in=in1) dataset.erc3_data_permno;
        by firmq;
        if(in1);
    run;


*------------------------------*;
*** get historic cusips and exchange codes;
    %macro cusip;   * dx.names ends at 30 june 2014, so that is last match;

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

        proc sql;
            create table cusips as
            select a.*, b.ncusip as cusip,  b.shrcd, b.exchcd, b.primexch
            from eadate a, dx.names b
            where   a.permno = b.permno
                and (b.date <= a.eadate <= b.nameendt);
        quit;

        * eliminate dups and save to perm set;
        proc sort data= cusips  nodupkey out=dataset.erc3_data_cusips;
            by permno eadate;
        run;    *0 lost;

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

*** merge back;
    proc sql;
        create  table comp as
        select  a.*, b.cusip, b.shrcd
        from    comp as a left join dataset.erc3_data_cusips as b
        on      a.permno = b.permno
            and a.eadate = b.eadate
        order   by firmq;
    quit;


***-----------------------------------------------------------------***
*
*       only keep if have permno and cusip;
*
***-----------------------------------------------------------------***;

*** cut;
    data comp;
        set comp;
        if(permno > .);
        if(cusip ne '');
        attrib _all_ label = '';
    run;


***-----------------------------------------------------------------***
*
*       pull ibes eadate and ue;
*
***-----------------------------------------------------------------***;


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

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

        * get ibes actual;
        proc sql;
            create  table actu as
            select  a.*, b.anndats as eadate_ibes, b.value as eps
            from    cusip a left join ibes.actu_epsus b
            on      a.cusip = b.cusip
                and (a.datadate - 7) le b.pends le (a.datadate + 7)
                and b.pdicity = 'QTR'
                and b.measure = 'EPS'
```

```
                and b.value > .
                and b.curr_act = 'USD';
        quit;

        %crsp_day(actu, actu, eadate_ibes);

        proc sort data=actu     nodupkey;
            by cusip datadate;
        run;


        * grab consensus where available;
        proc sql;
            create  table ibes as
            select  a.*, b.medest, b.numest
            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.eadate_ibes > b.statpers > (a.eadate_ibes - 100)      /* only if consensus 100 days or newer */
                and b.statpers > .
            group   by a.cusip, a.datadate
            having  b.statpers = max(b.statpers);
        quit;

        proc sort data=ibes     nodupkey    out=dataset.erc3_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.eadate_ibes, b.eps, b.medest, b.numest
        from    comp as a left join dataset.erc3_data_ibes as b
        on      a.cusip = b.cusip
            and a.datadate = b.datadate
        order   by firmq;
    quit;


***-----------------------------------------------------------------***
*
*       only keep if eadate same and have medest and shrcd 10 11
*
***-----------------------------------------------------------------***;

*** cut;
    data comp;
        set comp;

        if(1990 <= year <= 2013);

        if(eadate_crsp = eadate_ibes);

        if(shrcd in (10 11));

        * these are necessary to calc UE.  drop here to reduce runtime;
        if(eps * medest > .);

        drop eadate_ibes shrcd;
        attrib _all_ label = '';
    run;


***-----------------------------------------------------------------***
*
*       get ff25 portfolio assignment
*
***-----------------------------------------------------------------***;

*** clean up the FF data.  only run this once;
*------------------------------*;
    %macro ffclean;

    *** clean up the MVE breakpoints data from FF website;
        data dataset.erc3_data_ffclean_mvb;
            set dataset.ff_mv_breakpoints;
            if(year > 1985);
            if(month = 6);          * only need annual sort data;

            * only need 5 groups;
            drop p5 p10 p15 p25 p30 p35 p45 p50 p55 p65 p70 p75 p85 p90 p95 p100;
            drop firms month;
        run;


    *** clean up the bv breakpoints data from FF website;
        data dataset.erc3_data_ffclean_bv;
            set dataset.ff_bv_breakpoints;
```

```
            if(year > 1985);

            * only need 5 groups;
            drop p5 p10 p15 p25 p30 p35 p45 p50 p55 p65 p70 p75 p85 p90 p95 p100;
        run;


    *** clean up returns data;
        data ffew;
            set dataset.ff25_returns_daily_ew;
            if(year >= 1989);
            retdate = mdy(month, date, year);
            drop month date year;
            format retdate date9.;
        run;

        proc transpose data=ffew out=temp1 ;
            by retdate;
        run;

        data ffvw;
            set dataset.ff25_returns_daily_vw;
            if(year >= 1989);
            retdate = mdy(month, date, year);
            drop month date year;
            format retdate date9.;
        run;

        proc transpose data=ffvw out=temp2 ;
            by retdate;
        run;

        proc sql;
            create  table temp3 as
            select  a.retdate, a._name_, a.col1/100 as portret_ew, b.col1/100 as portret_vw
            from    temp1 as a left join temp2 as b
            on      a.retdate = b.retdate
                and a._name_ = b._name_;
        quit;

        data dataset.erc3_data_ffclean_ffew;
            set temp3;
            if(_name_ = 'SMALL_LoBM') then port25 = 11;
            if(_name_ = 'ME1_BM2   ') then port25 = 12;
            if(_name_ = 'ME1_BM3   ') then port25 = 13;
            if(_name_ = 'ME1_BM4   ') then port25 = 14;
            if(_name_ = 'SMALL_HiBM') then port25 = 15;
            if(_name_ = 'ME2_BM1   ') then port25 = 21;
            if(_name_ = 'ME2_BM2   ') then port25 = 22;
            if(_name_ = 'ME2_BM3   ') then port25 = 23;
            if(_name_ = 'ME2_BM4   ') then port25 = 24;
            if(_name_ = 'ME2_BM5   ') then port25 = 25;
            if(_name_ = 'ME3_BM1   ') then port25 = 31;
            if(_name_ = 'ME3_BM2   ') then port25 = 32;
            if(_name_ = 'ME3_BM3   ') then port25 = 33;
            if(_name_ = 'ME3_BM4   ') then port25 = 34;
            if(_name_ = 'ME3_BM5   ') then port25 = 35;
            if(_name_ = 'ME4_BM1   ') then port25 = 41;
            if(_name_ = 'ME4_BM2   ') then port25 = 42;
            if(_name_ = 'ME4_BM3   ') then port25 = 43;
            if(_name_ = 'ME4_BM4   ') then port25 = 44;
            if(_name_ = 'ME4_BM5   ') then port25 = 45;
            if(_name_ = 'BIG_LoBM  ') then port25 = 51;
            if(_name_ = 'ME5_BM2   ') then port25 = 52;
            if(_name_ = 'ME5_BM3   ') then port25 = 53;
            if(_name_ = 'ME5_BM4   ') then port25 = 54;
            if(_name_ = 'BIG_HiBM  ') then port25 = 55;
        run;

    %mend ffclean;
*   %ffclean;
*-------------------------------*;




*-------------------------------*;
    %macro ffassign;

    *** setup matching dates.  june for mve, dec of previous year for bv;
        data comp2;
            set comp;
            mve_ff_year = year(eadate_crsp);
            if(month(eadate_crsp) < 7) then mve_ff_year=mve_ff_year-1;
            btm_ff_year = mve_ff_year - 1;
            keep firmq gvkey permno datadate eadate_crsp mve_ff_year btm_ff_year;
        run;


    *** get firm's mve as of last day of july in preceding july;
        proc sql;
            create  table comp2 as
            select  a.*, (b.shares * abs(b.prc))/1000 as mve_ff, b.caldt
            from    comp2 as a left join mz.tsdata as b
            on      a.permno = b.permno
                and a.mve_ff_year = year(b.caldt)
```

```
                and month(b.caldt) = 6;
        quit;

        proc sql;
            create  table comp2 as
            select  a.*, b.p20, b.p40, b.p60, b.p80
            from    comp2 as a left join dataset.erc3_data_ffclean_mvb as b
            on      a.mve_ff_year = b.year;
        quit;

        data comp2;
            set comp2;
            if(mve_ff > .)      then mve_port = 1;
            if(mve_ff > p20)    then mve_port = 2;
            if(mve_ff > p40)    then mve_port = 3;
            if(mve_ff > p60)    then mve_port = 4;
            if(mve_ff > p80)    then mve_port = 5;
            drop p20 p40 p60 p80;
        run;


    *** get firm's btm as of last yearend of previous year;

        * mve at ye;
        proc sql;
            create  table comp3 as
            select  a.*, (b.shares * abs(b.prc))/1000 as mve_ye, b.caldt
            from    comp2 as a left join mz.tsdata as b
            on      a.permno = b.permno
                and a.btm_ff_year = year(b.caldt)
                and month(b.caldt) = 12;
        quit;

        * btm at ye, keeping only last in year;
        proc sql;
            create  table comp4 as
            select  a.*, b.seq, b.ceq, b.datadate as bvdate
            from    comp3 as a left join comp.funda as b
            on      a.gvkey = b.gvkey
                and a.btm_ff_year = year(b.datadate)
                and ((indfmt  = 'INDL') and (consol = 'C') and (popsrc = 'D') and (datafmt = 'STD'))
                and (b.ceq > . or b.seq > .)
            order   by gvkey, eadate_crsp, b.datadate descending;
        quit;

        proc sort data=comp4    nodupkey;
            by gvkey eadate_crsp;
        run;

        * calculate btm;
        data comp4;
            set comp4;
            btm_ff = seq / mve_ye;
            if(btm_ff = .) then btm_ff = ceq / mve_ye;
        run;

        *get breakpoints;
        proc sql;
            create  table comp4 as
            select  a.*, b.p20, b.p40, b.p60, b.p80
            from    comp4 as a left join dataset.erc3_data_ffclean_bv as b
            on      a.mve_ff_year = b.year;
        quit;

    *** assign portfolio and save;
        data dataset.erc3_data_ffassign;
            set comp4;
            if(btm_ff > .)      then btm_port = 1;
            if(btm_ff > p20)    then btm_port = 2;
            if(btm_ff > p40)    then btm_port = 3;
            if(btm_ff > p60)    then btm_port = 4;
            if(btm_ff > p80)    then btm_port = 5;
            ffport25            = mve_port * 10 + btm_port;
            drop p20 p40 p60 p80;
            keep firmq btm_port mve_port ffport25;
        run;

    %mend ffassign;
*   %ffassign;
*------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.ffport25
        from    comp as a left join dataset.erc3_data_ffassign as b
        on      a.firmq = b.firmq;
    quit;



***-----------------------------------------------------------------***
*
*       get car and pead
*
***-----------------------------------------------------------------***;
```

```
*-------------------------------*;
*** get portfolio-adjusted car based on ff25;
    %macro car_portff;

    *** get list events;
        proc sort data=comp nodupkey    out=permno (keep= permno eadate_crsp ffport25);
            where(ffport25 > .);
            by permno eadate_crsp;
        run;


    *** car calculation;

        *** get firm returns and put back;
            %event ( permno, permno2, eadate_crsp, 0, 2, vwretd, need_all= n, err_file =  );

            proc sql;
                create  table port2 as
                select  a.*, b.ret, b.caldt
                from    permno as a left join permno2 as b
                on      a.permno = b.permno
                    and a.eadate_crsp = b.eadate_crsp
                order   by permno, eadate_crsp, caldt;
            quit;

        *** get portfolio returns;
            proc sql;
                create  table port3 as
                select  a.*, b.portret_ew, b.portret_vw
                from    port2 as a left join dataset.erc3_data_ffclean_ffew as b
                on      a.ffport25 = b.port25
                    and a.caldt = b.retdate
                order   by permno, eadate_crsp, caldt;
            run;

        *** calculate abnormal return;
            proc sql;
                create  table dataset.erc3_data_car_portff as
                select  permno, eadate_crsp, count(ret) as n,
                        exp(sum(log(1+ret))) - exp(sum(log(1+portret_vw))) as car_portff_vw
                from    port3
                group   by permno, eadate_crsp
                having  n ge 3
                order   by permno, eadate_crsp
                ;
            quit;


    *** pead calculation;

        *** get firm returns and put back;
            %event ( permno, permno2, eadate_crsp, -3, 75, vwretd, need_all= n, err_file =  );

            proc sql;
                create  table port2 as
                select  a.*, b.ret, b.caldt
                from    permno as a left join permno2 as b
                on      a.permno = b.permno
                    and a.eadate_crsp = b.eadate_crsp
                order   by permno, eadate_crsp, caldt;
            quit;

        *** get portfolio returns;
            proc sql;
                create  table port3 as
                select  a.*, b.portret_ew, b.portret_vw
                from    port2 as a left join dataset.erc3_data_ffclean_ffew as b
                on      a.ffport25 = b.port25
                    and a.caldt = b.retdate
                order   by permno, eadate_crsp, caldt;
            run;

        *** calculate abnormal return;
            proc sql;
                create  table dataset.erc3_data_car_portff2 as
                select  permno, eadate_crsp, count(ret) as n,
                        exp(sum(log(1+ret))) - exp(sum(log(1+portret_vw))) as pead_portff_vw
                from    port3
                group   by permno, eadate_crsp
                having  n ge 3
                order   by permno, eadate_crsp
                ;
            quit;

    %mend car_portff;
*   %car_portff;
*-------------------------------*;
*** put back;
    proc sql;
        create  table comp as
        select  a.*,  b.car_portff_vw
        from    comp as a left join dataset.erc3_data_car_portff as b
        on      a.permno = b.permno
            and a.eadate_crsp = b.eadate_crsp;
```

```
    quit;

    proc sql;
        create  table comp as
        select  a.*, b.pead_portff_vw
        from    comp as a left join dataset.erc3_data_car_portff2 as b
        on      a.permno = b.permno
            and a.eadate_crsp = b.eadate_crsp;
    quit;


*------------------------------*;
*** abnormal volume;
    %macro vol;

    *------------------------------*;
    *** make a submacro just for the mkt;
        %macro vol_mkt;

            * pull crsp data;
            data crsp;
                set dx.tsdata (where=(1989 <= year(caldt) <= 2015)) ;
                if(vol * shares * prc > .);
                turn = vol / (shares * 100);
                dvol = log ( (vol * abs(prc)) + 1);
                keep caldt permno turn dvol ;
                attrib _all_ label = '';
            run;

            * calculate three-day average per firm days 0, 1, 2;
            proc sort data=crsp;
                by permno descending caldt;
            run;

            proc expand data = crsp out = crsp2;
                convert turn = turn_avg  / transformout=( movave 3);
                convert dvol = dvol_avg  / transformout=( movave 3);
                by permno;

            run;

            * calculate trailing avg;
            proc sort data=crsp2;
                by permno caldt;
            run;

            proc expand data = crsp2 out = crsp3;
                convert turn = turn_tavg  / transformout=( movave 70 lag 5);
                convert dvol = dvol_tavg  / transformout=( movave 70 lag 5);
                by permno;
            run;

            * make abnormal per firm;
            data crsp3;
                set crsp3;
                turn_abn = turn_avg - turn_tavg;
                dvol_abn = dvol_avg - dvol_tavg;
                if(turn_abn * dvol_abn > .);
            run;

            * make daily average;
            proc sort data=crsp3;
                by caldt;
            run;

            proc means data=crsp3   noprint;
                by caldt;
                var turn_abn;
                output out=dataset.erc3_data_abnvol_mkt     (drop= _:)
                                mean(turn_abn dvol_abn) = turn_abn_mkt dvol_abn_mkt;
            run;

        %mend vol_mkt;
        %vol_mkt;
    *------------------------------*;


    *** now get firm vol;

        * get list of permnos & eadates to get for;
        proc sort data=comp (keep= permno eadate_crsp where=(eadate_crsp > .))  nodupkey    out=eadate;
            by permno eadate_crsp;
        run;

        * pull crsp info;
        %event ( eadate, eadate2, eadate_crsp, 0, 2, vwretd, need_all= n, err_file = , ex_vars = vol shares prc );

        * calculate turnover and dollar volume per day;
        data eadate2;
            set eadate2;
            turn = vol / (shares * 1000);
            dvol = log( (vol * abs(prc)) + 1);
        run;

        proc sql;
            create  table eadate3 as
```

```
            select  distinct permno, eadate_crsp,
                    n(dvol) as n1, sum(dvol) / n(dvol) as dvol,
                    n(turn) as n2, sum(turn) / n(turn) as turn

            from    eadate2
            group   by permno, eadate_crsp
            having  n1 >= 3 and n2 >= 3;
        quit;


    *** calculate trailing average over 70 days ending with -5;
        %event ( eadate3, temp, eadate_crsp, 70, -5, vwretd, need_all= n, err_file = , ex_vars = vol shares prc );

        data temp;
            set temp;
            turn = vol / (shares * 1000);
            dvol = log( (vol * abs(prc)) + 1);
        run;

        proc sql;
            create  table temp2 as
            select  distinct permno, eadate_crsp,
                    n(dvol) as n1_tavg, sum(dvol) / n(dvol) as dvol_tavg,
                    n(turn) as n2_tavg, sum(turn) / n(turn) as turn_tavg
            from    temp
            group   by permno, eadate_crsp
            having  n1_tavg >= 20 and n2_tavg >= 20;
        quit;


    *** calculate firm-specific abnormal;
        proc sql;
            create  table eadate4 as
            select  a.permno, a.eadate_crsp, a.turn,
                    a.turn - b.turn_tavg as turn_abn
            from    eadate3 as a left join temp2 as b
            on      a.permno = b.permno
                and a.eadate_crsp = b.eadate_crsp
            order   by permno, eadate_crsp;
        quit;


    *** do firm & market abnormal;
        proc sql;
            create  table dataset.erc3_data_vol as
            select  a.*, b.turn_abn_mkt
            from    eadate4 as a left join dataset.erc3_data_abnvol_mkt as b
            on      a.eadate_crsp = b.caldt;
        quit;

    %mend vol;
*   %vol;
*-------------------------------*;

*** bring back in;
    proc sql;
        create  table comp as
        select  a.*, b.turn_abn, b.turn_abn_mkt
        from    comp as a left join dataset.erc3_data_vol as b
        on      a.permno = b.permno
            and a.eadate_crsp = b.eadate_crsp
        order   by firmq;
    quit;


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


*-------------------------------*;
    %macro weather;

    *** get unique list;
        proc sort data=comp nodupkey    out=dates (keep=eadate_crsp win2_end win_beg);
            by eadate_crsp;
        run;


    *** get weather;
        proc sql;
            create  table dates2 as
            select  a.*, b.recdate, b.cloud_mean, b.rain_mean, b.wind_mean
            from    dates as a left join dataset.weather_final_ny_v2_20150604 as b
            on      a.win_beg <= b.recdate <= a.win2_end
            order   by eadate_crsp, recdate;
        quit;

        proc means data=dates2  noprint;
            by eadate_crsp;
            var cloud_mean;
            output out=dates3   (drop= _:)
                                mean(cloud_mean rain_mean   wind_mean   )
                                =    cloud      rain        wind        ;
```

```
        run;

        data dataset.erc3_data_weather;
            set dates3;
            cloud_log       = log(cloud + 1);
            rain_log        = log(rain + 1);
            wind_log        = log(wind + 1);
        run;

    %mend weather;
*   %weather;
*------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.cloud_log, b.rain_log, b.wind_log
        from    comp as a left join dataset.erc3_data_weather as b
        on      a.eadate_crsp = b.eadate_crsp;
    quit;




*** get extreme weather for nyc;
*------------------------------*;
    %macro xtrm;

    *** get unique lists of dates;
        proc sort data=comp nodupkey    out=dates_erc (keep=eadate_crsp win2_end);
            by eadate_crsp;
        run;

        proc sort data=comp nodupkey    out=dates_pead (keep=eadate_crsp win2_end win75_end);
            by eadate_crsp;
        run;


    *** setup the weather data;
        data xtrm;
            set dataset.storm_events_final;
            if(event_type in ('Rip Current' 'Waterspout'))      then    delete;
            if(state = 'NY');
            if(county in ('New York' ));
            *if(county in ('New York' 'Queens' 'Bronx' 'Kings' 'Richmond'));

            xtrm_any = 1;

            * consolidate others;
            if(event_type in ('Flash Flood' 'Flood'))           then    xtrm_flood = 1;
                else                                                    xtrm_flood = 0;
            if(event_type in ('Thunderstorm Wind' 'Lightning')) then    xtrm_storm = 1;
                else                                                    xtrm_storm = 0;
            if(event_type in ('Funnel Cloud' 'Tornado'))        then    xtrm_torn = 1;
                else                                                    xtrm_torn = 0;
            if(event_type in ('Hail'))                          then    xtrm_hail = 1;
                else                                                    xtrm_hail = 0;
            if(event_type in ('Heavy Rain'))                    then    xtrm_rain = 1;
                else                                                    xtrm_rain = 0;
            if(xtrm_flood = xtrm_storm = xtrm_torn = xtrm_hail = xtrm_rain = 0) then    xtrm_other = 1;
                else                                                    xtrm_other = 0;

            keep date xtrm_flood xtrm_storm xtrm_torn xtrm_hail xtrm_rain xtrm_other xtrm_any;
        run;


    *** get and summarize erc window;
        proc sql;
            create  table xtrm_erc as
            select  a.*, b.*
            from    dates_erc as a, xtrm as b
            having  a.eadate_crsp <= b.date <= a.win2_end
            order   by eadate_crsp, date;
        quit;

        proc means data=xtrm_erc    noprint;
            by eadate_crsp;
            var xtrm_any;
            output out=dataset.erc3_data_xtrm_erc       (drop= _type_ _freq_)
                                    max(xtrm_flood xtrm_storm xtrm_torn xtrm_hail xtrm_rain xtrm_other xtrm_any)
                                    =   xtrm_flood xtrm_storm xtrm_torn xtrm_hail xtrm_rain xtrm_other xtrm_any;
        run;

    %mend xtrm;
*   %xtrm;
*------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.xtrm_any
        from    comp as a left join dataset.erc3_data_xtrm_erc as b
        on      a.eadate_crsp = b.eadate_crsp;
    quit;
```

```

***-----------------------------------------------------------------***
*
*       get avg analyst weather;
*       this uses the final output from code file A2c;
*
***-----------------------------------------------------------------***;

*-------------------------------*;
    %macro wf_anal;

        proc sql;
            create  table temp as
            select  distinct a.firmq, mean(b.wf3d) as wf_anal,
                    mean(b.xtrm_any3d) as xtrm_anal, n(b.wf3d) as wf_anal_n
            from    comp as a left join dataset.pull_analyst_pb_data_final as b
            on      a.gvkey = b.gvkey
                and a.eadate = b.rdq
            group   by a.firmq;
        quit;

        data temp;
            set temp;
            if(wf_anal_n <= 1) then do;
                wf_anal_n = .;
                wf_anal = .;
                xtrm_anal = .;
            end;
        run;

        * restandardize wf since no longer is mean/sd 0/1.  shoud not effect test statistics;
        proc standard data=temp mean=0 std=1    out=dataset.erc3_data_wf_anal;
            var wf_anal ;
        run;

    %mend wf_anal;
*   %wf_anal;
*-------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.wf_anal, b.xtrm_anal
        from    comp as a left join dataset.erc3_data_wf_anal as b
        on      a.firmq = b.firmq;
    quit;


***-----------------------------------------------------------------***
*
*       make eadate busyness
*
***-----------------------------------------------------------------***;

*-------------------------------*;
    %macro busy;

        * pull data;
        proc sql;
            create  table busy as
            select  a.gvkey, a.rdq, year(a.rdq) as year, week(a.rdq) as week
            from    comp.fundq a
            where   ((indfmt  = 'INDL') and (consol = 'C') and (popsrc = 'D') and (datafmt = 'STD'))
                and 1990 <= year(a.rdq) <= 2014
                and a.gvkey ne '';
        quit;

        proc sort data=busy nodupkey;
            by rdq gvkey;
        run;

        data dataset.erc3_data_busy;
            set busy (keep= rdq);
                by rdq;
            retain eaperday;
            if(first.rdq) then eaperday = 0;
            eaperday + 1;
            if(last.rdq) then output;
        run;

    %mend busy;
*   %busy;
*-------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.eaperday
        from    comp as a left join dataset.erc3_data_busy as b
        on      a.eadate = b.rdq;
    quit;


***-----------------------------------------------------------------***
*
*       get persist and volatility
```

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

*------------------------------*;
    %macro persist;

    *** get dataset;
        proc sql;
            create  table comp2 as
            select  a.firmq, a.gvkey, a.cshoq, (b.fyearq * 10 + b.fqtr) as fyearqtr, b.ibq / a.cshoq as ibqps
            from    comp as a left join comp.fundq as b
            on      a.gvkey = b.gvkey
                and a.fyearqtr > (b.fyearq * 10 + b.fqtr) >= (a.fyearqtr - 40)
            order   by firmq, fyearqtr, ibqps descending;
        quit;

        proc sort data=comp2    nodupkey;
            by firmq fyearqtr;
        run;

    *** get fourth quarter lag;
        proc sql;
            create  table comp3 as
            select  a.*, b.ibq / a.cshoq as ibqps_py, (b.fyearq * 10 + b.fqtr) as fyearqtr_py,
                    a.ibqps - (b.ibq / a.cshoq) as ibqps_diff
            from    comp2 as a left join comp.fundq as b
            on      a.gvkey = b.gvkey
                and (a.fyearqtr - 10) = (b.fyearq * 10 + b.fqtr)
            order   by firmq, fyearqtr, ibqps_py descending;
        quit;

        proc sort data=comp3    nodupkey;
            by firmq fyearqtr;
        run;


    *** calculat persist;
        options nonotes;
        proc reg data=comp3 noprint outest=beta edf;
            model ibqps = ibqps_py;
            by firmq;
        run;
        options notes;

        data dataset.erc3_data_persist;
            set beta;
            rename ibqps_py = persist;
            if(_edf_ >= 2);
            drop _: intercept ibqps;
        run;


    *** calculate volatility;
        proc means data=comp3   noprint;
            by firmq;
            var ibqps;
            output out=vol  stddev(ibqps_diff) = volatility n(ibqps_diff) = nobs;
        run;

        data dataset.erc3_data_volatility;
            set vol;
            if(nobs >= 4);
            keep firmq volatility;
        run;

    %mend persist;
*   %persist
*------------------------------*;

*** put back;
    proc sql;
        create  table comp as
        select  a.*, b.persist
        from    comp as a left join dataset.erc3_data_persist as b
        on      a.firmq = b.firmq;
    quit;

    proc sql;
        create  table comp as
        select  a.*, b.volatility
        from    comp as a left join dataset.erc3_data_volatility as b
        on      a.firmq = b.firmq;
    quit;



***-----------------------------------------------------------------***
*
*           get institutional ownership;
*
***-----------------------------------------------------------------***;


*--------------------------*;
    %macro instown;

```

```
        *** get a list of unique cusips in my dataset, then grab tfn data ;
            proc sort data=comp     nodupkey    out=temp (keep=cusip);
                by cusip;
                where(cusip ne '');
            run;

            proc sql;
                create  table s1 as
                select  a.*, b.mgrno, b.rdate, b.fdate, b.shares
                from    temp as a, tfn.s34 as b
                having  a.cusip = b.cusip
                    and year(b.rdate) ge 1985
                    and (b.shares > .);
            quit;


        *** sometimes there are multiple mgrno and rdate combinations due to multiple fdates.  just keep first;
            proc sort data=s1   nodupkey;
                by cusip mgrno rdate descending fdate;
            run;

            proc sort data=s1   nodupkey;
                by cusip mgrno rdate;
            run;


        *** consolidate by cusip and rdate;
            proc sql;
                create  table s2 as
                select  cusip, rdate, (sum(shares)) / 1000000 as shares_io
                from    s1
                group   by cusip, rdate;
            quit;


        *** match to datadate - make sure within 185 days;
            proc sort data=comp nodupkey    out=temp (keep=cusip datadate cshoq);
                by cusip datadate;
                where(cusip ne '');
            run;

            proc sql;
                create  table s3 as
                select  a.*, b.rdate, b.shares_io, b.shares_io / a.cshoq as instown
                from    temp as a left join s2 as b
                on      a.cusip = b.cusip
                    and ((a.datadate - 185) <= b.rdate <= a.datadate);
            quit;

            proc sort data=s3;
                by cusip datadate descending rdate;
            run;

            proc sort data=s3   nodupkey    out=dataset.erc3_data_instown;
                by cusip datadate;
            run;

    %mend instown;
*   %instown;
*-------------------------*;

*** merge back;
    proc sql;
        create  table comp as
        select  a.*, b.instown
        from    comp as a left join dataset.erc3_data_instown as b
        on      a.cusip = b.cusip
            and a.datadate = b.datadate;
    quit;



***-------------------------------------------------------------***;
*
*       calculate beta
*
*-------------------------------------------------------------***;


*-------------------------*;
    %macro beta;

        proc sort data=comp nodupkey    out=permno (keep= permno eadate_crsp);
            by permno eadate_crsp;
        run;

        %event ( permno, permno2, eadate_crsp, 252, -5, vwretd, need_all= n );

        options nonotes;
        proc reg data=permno2 noprint outest=permno3 edf;
            model ret= vwretd;
            by permno eadate_crsp;
        run;
        options notes;

        * drop obs with insufficient data for reg;
```

```
        data dataset.erc3_data_beta (keep= permno eadate_crsp beta);
            set permno3;
            if(_EDF_ < 20) then delete;
            rename vwretd= beta;
        run;

    %mend beta;
*   %beta;
*-------------------------*;

*** bring back;
    proc sql;
        create  table comp as
        select  a.*, b.beta
        from    comp as a left join dataset.erc3_data_beta as b
        on      a.permno = b.permno
            and a.eadate_crsp = b.eadate_crsp;
    quit;




***-----------------------------------------------------------------***
*
*       make vars;
*
***-----------------------------------------------------------------***;

*** make;
    data comp;
        set comp;

        * rescale market measures;
        car_portff_vw       = car_portff_vw * 100;
        pead_portff_vw      = pead_portff_vw * 100;
        turn_abn            = turn_abn * 100;

        * make other variables;
        ue                  = (eps - medest) / prccq;
        mve                 = cshoq * prccq;
        if(mve > 0)         then size = log(mve);
        if(mve > 0)         then btm = seqq / mve;
        replag              = eadate - datadate;
        if(replag > 0)      then replag_log = log(replag);
        numest_log          = log(numest);
        fq4                 = 0;
        if(fqtr = 4)        then fq4 = 1;
        loss                = 0;
        if(eps < 0)         then loss = 1;
        if(instown > 1) then instown = 1;
        miss                = 0;
        if(ue < 0)          then miss = 1;

        dow                 = weekday(eadate);
        friday              = 0;
        if(dow = 6)         then friday = 1;

        if(xtrm_any = .)    then xtrm_any = 0;

    run;



***-----------------------------------------------------------------***
*
*       cut;
*
***-----------------------------------------------------------------***;

*** cut;
    data comp;
        set comp;

        if(1990 <= year <= 2013);
        if(numest > 1);                 * if numest based on fewer than 2 analysts then UE is invalid;

        * must have market measures;
        if(turn_abn > .);
        if(car_portff_vw * pead_portff_vw > .);

        * outliers and likely data errors;
        if(prccq >= 1 and ue < 1);
        if(0 < replag < 365);

        * necessary regression variables;
        if(size * btm * replag_log * persist * volatility * beta * instown > .);
        if(cloud_log * rain_log * wind_log > .);
    run;


*** standardize cloud rain wind;
    proc standard data=comp     mean=0 std=1        out=comp;
        var rain_log wind_log cloud_log ;
    run;
```

```
*** make weather factor;
    PROC FACTOR DATA=comp noprint METHOD=principal rotate=v n=1 out=comp (rename=(factor1 = wf));
        var cloud_log rain_log wind_log;
    RUN;


*** make busy day variable;
    proc sort data=comp;
        by year;
    run;

    proc rank data=comp (keep= firmq    eaperday year)
            out=ranks   (drop=          eaperday year)
            groups = 10
            ties = low;
        by      year;
        var     eaperday;
        ranks   busy;
    run;

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


*** form UE decile ranks;
    proc rank data=comp (keep= firmq    ue )
            out=ranks   (drop=          ue )
            groups = 10
            ties = low;
        var     ue;
        ranks   ue_dec;
    run;

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

    proc standard data=comp     mean=0      out=comp;
        var zue_dec;
    run;


*** de-mean and interact the UE controls;
    %macro std(var);
        data comp;
            set comp;
            zz&var = &var;
        run;

        proc standard data=comp     mean=0      out=comp;
            var zz&var;
        run;

        data comp;
            set comp;
            ue_dec_z&var = zue_dec * zz&var;
*           drop zz&var;
        run;

    %mend std;

    %std(wf);
    %std(wf_anal);
    %std(xtrm_any);
    %std(xtrm_anal);
    %std(size);
    %std(replag_log);
    %std(btm);
    %std(persist);
    %std(volatility);
    %std(instown);
    %std(beta);
    %std(numest_log);
    %std(loss);
    %std(fq4);
    %std(friday);
    %std(busy);



*** clean up the dataset;
    data comp;
        set comp;

        rename  ue_dec_zwf      = ue_dec_wf;

        drop cusip win_beg win2_end medest  atq cshoq  seqq eadate_crsp prccq
            eps datadate comp_zip ibq  zz:
            win75_end mergeall eaperday
```

```
            ffport25
            cloud_log rain_log wind_log
            mve replag numest  numest   ;

        attrib _all_ label = '';

    run;



*** winsor;
    %winsor(dsetin=comp, dsetout=comp, byvar=none,
            vars=   ue
                    car_portff_vw   pead_portff_vw
                    turn_abn        turn_abn_mkt

                    size            ue_dec_zsize
                    replag_log      ue_dec_zreplag_log
                    btm             ue_dec_zbtm
                    numest_log      ue_dec_znumest_log
                    persist         ue_dec_zpersist
                    volatility      ue_dec_zvolatility
                    beta            ue_dec_zbeta
                    instown         ue_dec_zinstown
                                    ue_dec_zfriday
                                    ue_dec_zloss
                                    ue_dec_zfq4
                                    ue_dec_zbusy

                    wf
                    wf_anal
                    ue_dec_wf
                    ue_dec_zwf_anal
                    ue_dec_zxtrm_any
                    ue_dec_zxtrm_anal

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


***-----------------------------------------------------------------***
*
*       save;
*
***-----------------------------------------------------------------***;

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


*** export;
    data comp;
        set comp;
    run;

    proc export data=comp outfile= "Market_Data_Final.dta" replace;


*** save a empty dataset for JAR;
    data this.M4_observations;
        set comp;
        keep gvkey eadate fyearqtr;
    run;

endsas;
```