```
*** file A2b;

libname this 'YOUR_DIRECTORY';




***-----------------------------------------------------------------***
*
*       create a list of stations within 50 miles of each zipcode appearing in the sample;
*
***-----------------------------------------------------------------***;

*-------------------------------*;
%macro substation;

*** this list of zip codes was generated by an older version of what is now A2c;
    data zip_codes;
        infile "YOUR_DIRECTORY/zip_codes.csv" dsd delimiter=',' firstobs=2 missover;
        length city $40.;
        input   statex  $
                city   $
                zipx $
                lat
                long
            ;
        state = substr(statex, 1, 2);

        if length(zipx) >= 5 then zip = substr(zipx, 1, 5);
        else zip = repeat('0', 5 - length(zipx) - 1) || zipx;

        drop statex zipx city ;
        mergeall = 1;
    run;    *141 zip codes;

    proc sort data=zip_codes    nodupkey;
        by zip;
    run;


*** bring in the list of weather station locations downloaded from the isd-lite database;
    data history;
        infile "YOUR_DIRECTORY/isd-history.csv" dsd delimiter=',' firstobs=2 missover lrecl=300;
        length name $40.;
        input
                usaf  $
                wban  $
                name  $
                country $
                state $
                icao  $
                lat
                long
                elevation
            ;
        mergeall = 1;
        if(lat < 0) then delete;
        drop country elevation;
    run;


*** merge the two together. ;
    proc sql;
        create  table stations as
        select  a.*, b.usaf, b.wban, b.name, geodist(a.lat, a.long, b.lat, b.long, 'M') as distance
        from    zip_codes as a left join history as b
        on      a.mergeall = b.mergeall
        order   by zip, distance;
    quit;


*** drop all stations more than 50 miles away;
    data this.stations_anal_insample;
        set stations;
        if(distance > 50) then delete;
    run;

%mend substation;
*%substation;
*-------------------------------*;



***-----------------------------------------------------------------***
*
*       read and clean the weather data. ;
*
***-----------------------------------------------------------------***;

*-------------------------------*;
%macro read;

*** get stations;
    data stations;
        set this.stations_anal_insample;
        if(state in ('WA' 'OR' 'CA' 'NV'))                      then tzone = 'west';
```

```
        if(state in ('MT' 'ID' 'WY' 'UT' 'CO' 'AZ' 'NM'))       then tzone = 'moun';
        if(state in ('AK'))                                      then tzone = 'alas';
        if(state in ('HI'))                                      then tzone = 'hawi';
        if(state in ('ND' 'MN' 'WI' 'SD' 'NE' 'IA' 'IL' 'KS' 'MO' 'OK' 'AR' 'TN' 'TX' 'LA' 'AR' 'MS' 'AL')) then tzone = 'cent';
        if(tzone = '')                                           then tzone = 'east';
        if(usaf ne '' and wban ne '') then haveloc =1 ;          * checked and all have locations;
    run;

    proc sort data=stations nodupkey;
        by usaf wban;
    run;


*** read in the weather data;
    data weather;
        infile "YOUR_DIRECTORY/weather_data_1990-2014.txt"
            dsd delimiter='09'x firstobs=2 missover;
        input   USAF   $
                WBAN   $
                year
                month
                day
                hour
                temp
                dewpoint
                pressure
                winddir
                windvelo
                cloudy
                precip1h
                precip6h
            ;

        recdate_utc = mdy(month, day, year);
        recdatetime_utc = dhms(recdate_utc, hour, 0, 0);
        format  recdatetime_utc datetime.;

        * this deletes 9 and 10 for days when cloudiness is obscured;
        if(cloudy < 0 or cloudy > 8) then cloudy = .;

        * set trace precipitation to 1, which is .1mm;
        if(precip1h = -1) then precip1h = 1;

        * drop obs with all missing main variables;
        if(temp = windvelo = cloudy = precip1h = .) then delete;

        * delete missing year or after 2005 (the last possible year in the analyst sample);
        if(. < year <= 2005);

        drop precip6h dewpoint pressure winddir month day year recdate_utc hour;
    run;


*** only keep ones on station list;
    proc sql;
        create  table this.weather2_anal as
        select  b.*, a.tzone
        from    stations as a, weather as b
        having  a.usaf = b.usaf
            and a.wban = b.wban;
    quit;

%mend read;
*%read;
*------------------------------*;




*------------------------------*;

%macro read2;

*** fix times and weather measurements;
    data this.weather3_anal;
        set this.weather2_anal;

        * make time variable;
        if(tzone='east') then recdatetime = intnx('hour', recdatetime_utc, -5);
        if(tzone='cent') then recdatetime = intnx('hour', recdatetime_utc, -6);
        if(tzone='moun') then recdatetime = intnx('hour', recdatetime_utc, -7);
        if(tzone='west') then recdatetime = intnx('hour', recdatetime_utc, -8);
        if(tzone='alas') then recdatetime = intnx('hour', recdatetime_utc, -9);
        if(tzone='hawi') then recdatetime = intnx('hour', recdatetime_utc, -11);

        * fix daylight savings time;
        if( ('27APR86'd le datepart(recdatetime) le '26OCT86'd) or
            ('05APR87'd le datepart(recdatetime) le '25OCT87'd) or
            ('03APR88'd le datepart(recdatetime) le '30OCT88'd) or
            ('02APR89'd le datepart(recdatetime) le '29OCT89'd) or
            ('01APR90'd le datepart(recdatetime) le '28OCT90'd) or
            ('07APR91'd le datepart(recdatetime) le '27OCT91'd) or
            ('05APR92'd le datepart(recdatetime) le '25OCT92'd) or
            ('04APR93'd le datepart(recdatetime) le '31OCT93'd) or
            ('03APR94'd le datepart(recdatetime) le '30OCT94'd) or
            ('02APR95'd le datepart(recdatetime) le '29OCT95'd) or
```

```
            ('07APR96'd le datepart(recdatetime) le '27OCT96'd) or
            ('06APR97'd le datepart(recdatetime) le '26OCT97'd) or
            ('05APR98'd le datepart(recdatetime) le '25OCT98'd) or
            ('04APR99'd le datepart(recdatetime) le '31OCT99'd) or
            ('02APR00'd le datepart(recdatetime) le '29OCT00'd) or
            ('01APR01'd le datepart(recdatetime) le '28OCT01'd) or
            ('07APR02'd le datepart(recdatetime) le '27OCT02'd) or
            ('06APR03'd le datepart(recdatetime) le '26OCT03'd) or
            ('04APR04'd le datepart(recdatetime) le '31OCT04'd) or
            ('03APR05'd le datepart(recdatetime) le '30OCT05'd) or
            ('02APR06'd le datepart(recdatetime) le '29OCT06'd) or
            ('11MAR07'd le datepart(recdatetime) le '04NOV07'd) or
            ('09MAR08'd le datepart(recdatetime) le '02NOV08'd) or
            ('08MAR09'd le datepart(recdatetime) le '01NOV09'd) or
            ('14MAR10'd le datepart(recdatetime) le '07NOV10'd) or
            ('13MAR11'd le datepart(recdatetime) le '06NOV11'd) or
            ('11MAR12'd le datepart(recdatetime) le '04NOV12'd) or
            ('10MAR13'd le datepart(recdatetime) le '03NOV13'd) or
            ('09MAR14'd le datepart(recdatetime) le '02NOV14'd) )
            then  recdatetime = intnx('hour', recdatetime, +1);

        recdate = datepart(recdatetime);
        hour    = hour(recdatetime);

        format  recdatetime datetime.;
        format  recdate date9.;

        * keep only daytime 6am - 6pm;
        if(6 <= hour <= 17) ;

        * convert celcius (which is *10) into fareneit;
        temp        = ((temp/10)        * (9/5)) + 32;

        * rain is in mm*10.  change to mm;
        precip1h        = precip1h  / 10;

        * wind is in meters per second * 10.  change to mph;
        windvelo    = (windvelo     / 10) * 2.2369362920544;

        drop recdatetime_utc recdatetime tzone;
    run;

%mend read2;
*%read2;
*------------------------------*;



***-----------------------------------------------------------------***
*
*       summarize and save at the daily level;
*
***-----------------------------------------------------------------***;

*------------------------------*;
%macro summarize;

*** load and sort;
    proc sort data=this.weather3_anal       out=weather4_anal;
        by usaf wban recdate;
    run;


*** summarize and save;
    proc means data=weather4_anal   noprint     n;
        by usaf wban recdate;
        var cloudy;
        output  out = this.daycloud_trimmed_anal (drop=_type_ _freq_)
                mean(cloudy precip1h windvelo)  = cloud_mean    rain_mean   wind_mean
                n(cloudy precip1h windvelo)     = cloud_n       rain_n      wind_n;
    run;

%mend summarize;
*%summarize;
*------------------------------*;





***-----------------------------------------------------------------***
*
*       put together a time series of closest available cloud, rain, and wind for each obs
*
***-----------------------------------------------------------------***;

*------------------------------*;
%macro finalset;

*** make a time structure of all weather stations;

    * get a list of all recdates;
    proc sort data=this.daycloud_trimmed_anal   nodupkey    out=time (keep= recdate);
        by recdate;
    run;

```

```
    data time;
        set time;
        mergeall = 1;
    run;

    * get a list of all zip codes;
    proc sort data=this.stations_anal_insample  nodupkey    out=zip (keep= zip usaf wban distance);
        by zip usaf wban;
    run;

    data zip;
        set zip;
        mergeall=1;
    run;

    * put together to make a complete time structure with all stations;
    proc sql;
        create  table time2 (drop=mergeall) as
        select  a.zip, a.usaf, a.wban, a.distance, b.*
        from    zip as a left join time as b
        on      a.mergeall = b.mergeall
        order   by zip, recdate, distance, usaf, wban;
    quit;


*** make a master dataset with just zip, recdate;
    proc sort data=time2    nodupkey    out=master (drop= usaf wban distance);
        by zip recdate;
    run;


*** merge in the closest of each of four weather variables;
    %macro findweather(var);

        proc sql;
            create  table cloudy2 as
            select  a.*, b.&var._mean, b.&var._n
            from    time2 as a left join this.daycloud_trimmed_anal as b
            on      a.usaf = b.usaf
                and a.wban = b.wban
                and a.recdate = b.recdate
            order   by zip, recdate, distance;
        quit;

        * set the minimum count threshold and keep closest remaining;
        data cloudy2;
            set cloudy2;
            if(&var._n < 4) then delete;                                    /* drop if fewer than 4 hourly obs to average*/;
        run;

        proc sort data=cloudy2  nodupkey;
            by zip recdate;
        run;

        * merge into master;
        proc sql;
            create  table master as
            select  a.*, b.&var._mean
            from    master as a left join cloudy2 as b
            on      a.zip = b.zip
                and a.recdate = b.recdate;
        quit;

    %mend findweather;
    %findweather(cloud);
    %findweather(rain);
    %findweather(wind);


*** save the final series ;
    proc sort data=master   nodupkey;
        by zip recdate;
    run;

    data this.weather_final_anal_20160818;
        set master;
        if(1996 <= year(recdate) <= 2005);
    run;

%mend finalset;
*%finalset
*-------------------------------*;
```