

Paper ST-142

Creating County-Level Estimates from National Weather Service Data

Liang Wei, MS, MPH, Statistician, The Ginn Group/Northrop Grumman
Laurie Barker, MSPH, Mathematical Statistician, Division of Oral Health,
National Center for Chronic Disease Prevention and Health Promotion,
Centers for Disease Control and Prevention

ABSTRACT

An analysis of health survey data required estimates of daily temperature and precipitation for each day in 1999–2005 for ≈3,000 U.S. counties. The National Oceanic and Atmospheric Administration's National Climatic Data Center (NCDC) provides climatological data from the ≈6,000 National Weather Service (NWS) surface observing stations in the Surface Land Daily Cooperative Summary of the Day (DSI-3200) data set[1] and station location data in the Master Station File. Creating county estimates involved: 1) creating temperature and precipitation SAS® data files at the weather station level from 58,416 raw ASCII files, 2) creating county-level daily temperature and precipitation average based on the six nearest weather stations, and 3) predicting daily temperature and precipitation at each county centroid using ordinary kriging. We used SAS procedures DATA, SQL, SORT, APPEND and MERGE to read and extract the needed data; read many files with wildcards; transposed data sets from one record with many variables to multiple records and vice versa; conducted one-to-one and one-to-many merging; replaced variable values across data sets; computed the number of observations; and selected a fixed number of observations within by-group. We used the SAS procedure PROC MIXED to produce county estimates with ordinary kriging. We used SAS arrays and macros to make the complex coding more concise and processing more efficient. We provide SAS code and a description of the procedures to document the process used to obtain the county-level estimates as a reference for other analysts working with these or other large collections of data stored in many files.

INTRODUCTION

An analysis of health survey data for daily fluid intake required estimates of daily maximum temperature and precipitation for each day in 1999–2005 for 3,219 U.S. counties and county equivalents. The National Oceanic and Atmospheric Administration's National Climatic Data Center (NCDC) provides climatological data from the ≈6,000 National Weather Service (NWS) surface observing stations in the Surface Land Daily Cooperative Summary of the Day (DSI-3200) data set[1] and station location data in the Master Station File. Creating county estimates involved: 1) creating temperature and precipitation SAS data files at the weather station level from 58,416 raw ASCII files; 2) creating county-level estimates for maximum daily temperature and precipitation average based on six nearest weather stations; 3) predicting maximum daily temperature and precipitation at each county centroid using ordinary kriging; and 4) comparing estimates from the simple average and ordinary kriging methods. We describe the data management and analysis, providing examples of SAS code, and compare estimates from the two methods. The large number of observations and constraints on computing power required a balance between the most feasible and most sophisticated approaches to producing these estimates for all counties in the United States.

DATA MANAGEMENT

1) CREATE TEMPERATURE AND PRECIPITATION SAS DATA FILES AT WEATHER STATION LEVEL FROM RAW ASCII FILES

Step 1: Read ASCII text files into SAS data sets.

There are several methods in SAS to read multiple files. One method is to create an SAS macro to read one file, create an SAS system data table, append it to the master template, and process other remaining text files in the same pattern. In this method, the file name information is stored in a macro variable, and the value of the macro variable changes with each iteration of the loop. A shortcoming of this method is that the file names have to be typed in for each macro iteration loop.

Another method uses the FILEVAR=<column name> option for the INFILE statement to read the names of multiple files. For this method, an SAS data set containing a variable storing all qualified file names has to be set up first. A third method uses wildcards on the FILENAME and INFILE statements. With this method, the SAS data set containing raw ASCII file names would not need to be created. Therefore, this method makes reading multiple files

simple in comparison to the first two methods described. This method was used in reading 58,416 climate files from ASCII format.

The climate ASCII files for this project contain 7 years of data spanning 1999–2005, which were saved as seven zipped files, one per calendar year. Each year's raw text file was unzipped to 53 different file folders named by state abbreviation. Therefore, data were read in by year and were converted into 53 different SAS data sets, one per state, for each year.

The maximum temperature and precipitation data were extracted into separate data sets because one weather station may have maximum temperature available, but may not have precipitation data available. Only SAS code to create temperature estimates are provided here; however, the code for the precipitation estimates follows the same pattern.

The following SAS code will read temperature data into SAS data sets from all climate files in the 53 subdirectories of the 2004 folder into 53 SAS data sets:

```
1 libname path 'C:\NOAA\DATA\2004';
2 %macro mytransfer(data= );
3 DATA path.&data;
4 array DOM[62] DAY01-DAY62;
5 array HOUR[62] HOUR01-HOUR62;
6 array VALUE[62] VALUE01-VALUE62;
7 array QCFLGA[62] QCFLG1_01-QCFLG1_62;
8 array QCFLGB[62] QCFLG2_01-QCFLG2_62;
9 infile myfiles LRECL=774 TRUNCOVER;
10 input METELTYPE $ 12-15 @;
11 if METELTYPE="TMAX";
12 input @1 RECTYPE $ 3. STATIONID $ 8. @16 METELUNIT $ 2.
13 Year 4. MONTH 2. @28 NDATAPRT 3. @;
14 if NDATAPRT>0 then
15 do i= 1 to NDATAPRT;
16 x=((i-1)*12);
17 input @(31+x) DOM[i] 2. HOUR[i] 2. VALUE[i] 6.
18 QCFLGA[i] 1. QCFLGB[i] $ 1. @;
19 end;
20 output;
21 drop i x;
22 RUN;
23 %mend;
24 filename myfile 'C:\noaa\2004\AK\*. *';
25 %mytransfer(data=AK);
/*... repeat for other states ...*/
```

The first line of this SAS code creates an SAS LIBNAME to define a subdirectory for the multiple ASCII files that need to be transferred into SAS data sets. Lines 2–23 create an SAS macro that processes each climate ASCII data file containing the maximum temperature data along with other meteorological data. Within this SAS macro, line 2 creates a macro name, line 3 defines a macro data set, lines 4–8 define arrays to store information about days of the month, observed hours, temperature values, and quality control flags.

Line 9 is the INFILE statement, with the options of LRECL and TRUNCOVER. The LRECL option is used to tell SAS to expect records that are 774 characters long. The TRUNCOVER option assigns missing values to all variables when a short record is encountered rather than move to a new record to find more data. The INPUT statement on Line 10 specifies to read variable METELTYPE (Meteorological Element Type) (i.e., maximum temperature, minimum temperature, precipitation) at column 15. Lines 11–13 use one IF statement and one INPUT statement to tell SAS to read variables of STATIONID, METELUNIT, YEAR, MONTH, and NDATAPRT (Number of data elements in record) when value of variable METELTYPE is equal to "TMAX." These lines ensure SAS only reads information about maximum temperatures from the raw data.

Lines 14–19 use an IF statement and a DO loop to read each day's information about maximum temperature in every month. This information includes day of month (DOM), observed hours (HOUR), maximum temperature value (VALUE) and quality flags (QCFLGA and QCFLGB). The DO loop ensures SAS reads all days' information in from the

raw data. The OUTPUT statement on line 20 tells SAS to write variables into a temporary SAS data set. Line 21 uses a drop statement to tell SAS to drop the temporary counter variables “i” and “x.” Line 24 uses filename statement to define the subdirectory for the variable of “myfile” in the previous INFILE statement.

Line 25 is a macro run statement—after submitting this statement, SAS reads all files in the subdirectory defined by the FILENAME statement. For example, the first macro in the above example will read all raw data files in the subdirectory of 'C:\noaa\2004\AK\'. The wildcard of “*.” in the FILENAME statement will ensure that SAS reads all raw data files in the directory from the state of Alaska. By defining FILENAME statement using different subdirectories, which includes data files from other states and recalling SAS macro repeatedly, all raw data files from all states in one specific year can be read into the different SAS data sets.

Step 2: Clean data for the created SAS data sets.

Two parts of SAS code were used for this step. One part is used to convert each station's daily observed temperature (multiple variables in one record, maximum 62 temperature variables each month for each station) into multiple records (one record per day). Each station's record may contain both valid and invalid daily temperature values, because one original maximum temperature value and one edited maximum temperature value could be recorded for each day of the month. The goal is to keep one valid temperature value for each day of the month. After running this part of the SAS code, each station will have multiple records to store daily maximum temperature (one record for each day; maximum 365 records for non-leap year, 366 records for a leap year).

If you need to include source code:

```
1 libname path 'C:\NOAA\DATA\2004';
2 %macro onetomany;
3 data work.&data2(keep=stationid year month ndataprt temperature
4 ndays);
5 attrib
6 ndays format=2. informat=2. length=3 label="days with valid value"
7 temperature format=3. informat=3. length=3 label="valid
  temperature";
8 set path.&data1;
9 array days(62) day01-day62;
10 array values(62) value01-value62;
11 array qcflgb[62] qcflg2_01-qcflg2_62;
12 N=NDATAPRT;
13 do i=1 to N;
14   if qcflgb[i]^='2' then do;
15     temperature=values(i);
16     ndays=days(i);
17     output;
18   end;
19 end;
20 drop i N;
21 run;
22 %mend;
23 %let data1=ak;
24 %let data2=ak1;
25 %onetomany;
/*... repeat for other states ...*/
```

In the above SAS code, we start with a LIBNAME statement to define a subdirectory for the SAS data sets. Lines 2–22 define a macro to transfer data from one record into multiple records. In this macro, line 2 defines a macro name; lines 5–7 use an attrib statement to define formats and labels for the variables in a new data set. Lines 9–11 define arrays to store values from variables of day01–day62, value01–value62, and qcflg2_01–qcflg2_62. Line 12 creates a new counter variable N in the new data set from variable NDATAPRT in the old data set. Lines 13–18 use a DO loop to convert each station's single record containing multiple variables for day and temperature into one record per day per station.

After running the SAS code above, each station will have multiple records with one variable for daily maximum temperature. By calling the macro repeatedly (once per each year and state combination), temperature data from all

states will be converted into multiple records per station from one record per station.

The second part of this step is to convert each station's multiple temperature records into one record with multiple variables (one record for each station, one variable per day). This is accomplished by the "manytoone" macro:

```
1 %macro manytoone;
2 data work.&data2(keep=stationid year jan1-jan31 feb1-feb29 mar1-mar31 3 apr1-apr30
may1-may31 jun1-jun30
4     jul1-jul31 aug1-aug31 sep1-sep30 oct1-oct31 nov1-nov30 dec1-
5 dec31);
6 attrib
7     jan1-jan31 feb1-feb29 mar1-mar31 apr1-apr30 may1-may31 jun1-jun30
8     jul1-jul31 aug1-aug31 sep1-sep30 oct1-oct31 nov1-nov30 dec1-
9 dec31
10 format=3. informat=3. length=3;
11 retain jan1-jan31 feb1-feb29 mar1-mar31 apr1-apr30 may1-may31 jun1-12 jun30
12     jul1-jul31 aug1-aug31 sep1-sep30 oct1-oct31 nov1-nov30 dec1-
13 dec31;
14 array tmax(366) jan1-jan31 feb1-feb29 mar1-mar31 apr1-apr30 may1-
15 may31 jun1-jun30
16     jul1-jul31 aug1-aug31 sep1-sep30 oct1-oct31 nov1-nov30 dec1-
17 dec31;
18 array jan(31) jan1-jan31;
19 /*... repeat for other months ...*/
32 set work.&data1;
33 by stationid;
34 if first.stationid then do;
35 do i=1 to 366;
36 tmax(i)=.;
37 end;
38 end;
39 if month=1 then do;
40 m1=ndays;
41 jan(m1)=temperature;
42 end;
43 /*... repeat for other months ...*/
88 if last.stationid then output;
89 drop m1-m12;
90 run;
91 %mend manytoone;
92 %let data1=ak1;
93 %let data2=ak2;
94 %manytoone;
95 /*... repeat for other states ...*/
```

The SAS code above uses a macro to convert one station's multiple records into a simple record for one station. Line 1 defines an SAS macro. Lines 2–5 define a data set and tell SAS what variables need to be included in the new data set. Lines 6–10 use an ATTRIB statement to define formats and labels for variables representing daily maximum temperature (Jan1–Dec31). Lines 11–14 use a RETAIN statement to hold the values of variables across iterations of the data step. Lines 15–31 use arrays to store the daily maximum temperature values. Lines 32–33 tell SAS to transpose the old data set into a new one by Station ID. Lines 34–90 is a DO loop iteration that returns daily maximum temperature only for those months and days with valid observed maximum temperature values.

After calling the SAS macro above, one data set containing multiple variables for daily maximum temperature is created for each station within a state. The macro is repeated once per state to obtain data sets for all stations in every state.

Step 3: Combine data by state.

```
1 data path.tmax04;
2 attrib
```

```

3 STATIONID format=$8. informat=$8. length=$8 label="Station ID"
4 YEAR format=4. informat=4. length=4 label="Year"
5 jan1-jan31 feb1-feb29 mar1-mar31 apr1-apr30 may1-may31 jun1-jun30
6 jul1-jul31 aug1-aug31 sep1-sep30 oct1-oct31 nov1-nov30 dec1- dec31
7 format=3. informat=3. length=3;
8 set work.ak2;
9 run;

10 %macro append(data=);
11 proc append base=path.tmax04 data=work.&data;
12 run;
13 %mend append;
14 %append(data=AL2);
    /*... repeat for other states ...*/

```

The lines 1–9 of the SAS code above create a data set called “tmax04,” which includes variables containing weather station identification number, years of observations, and daily observed temperatures from data set “ak2” (maximum temperature data from Alaska). Lines 10–14 use a macro of SAS Proc Append to read in the same data from all other states.

Step 4: Merge temperature and precipitation data with weather station master file and Census county file.

The weather station master file contains station identification number, state abbreviation, county name and latitude and longitude. The Census county file [2] contains state abbreviation, county name, state and county Federal Information Processing Standard (FIPS) codes. The weather station file and Census county file were merged by state abbreviation and county name to create a new master file, then this new master file was merged with the temperature and precipitation data files. After these merges, the new files contain station identification, state FIPS, county FIPS, and daily observed maximum temperature and precipitation information ready for analysis. The SAS code for this part is fairly simple and was omitted here.

ESTIMATING MAXIMUM DAILY TEMPERATURE BY TWO METHODS

2) ESTIMATE MAXIMUM DAILY TEMPERATURE AND PRECIPITATION AS THE AVERAGE OF OBSERVATIONS FROM THE SIX NEAREST WEATHER STATIONS

Step 1: Set up temperature and precipitation data file in county-level based on neighboring state relationship.

The project required data for the years 1999–2005. The number of stations and data files varied by year and much of the data processing and estimation was handled by year. For ease of illustration, most examples use data from 2005. There are 6,169 weather stations with data files from 2005. These 6,169 weather stations are located in 2,775 counties or county equivalents. Among these 2,775 counties, 2,665 (95.6%) of them have six or fewer weather stations; there were no NWS data for 444 of the 3,219 counties or county equivalents in the United States. Observed daily temperature and precipitation values may be missing. The daily temperature and precipitation in these counties can be estimated by using weather stations located in neighboring counties. Constraints on processing power prevented estimation for all U.S. counties at once. Instead, estimates were prepared for each county within a single state. To avoid border effects, weather stations from neighboring states were included in the process of estimating temperatures in each county for each state. A state neighboring relationship data file including the state abbreviation of each state and each state’s neighboring states’ abbreviation was created manually. A new file was created by merging the state neighboring relationship file with Census county file. This new file is used to build a neighboring relationship between states and counties.

Through merging of a single state’s neighbor relationship file with temperature and precipitation data file at the weather station level, a new temperature or precipitation file is created containing observed values from weather stations in the domain state and neighboring states. The macro “merge_neighbor” performs this step:

```

%macro merge_neighbor;
%do i=1999 %to 2005;
proc sql;
create table vt&i as

```

```

select*
from temp&i(drop=state_fips nwsst statename county county_fips), county.neighbors1
where temp&i .state=neighbors1.state_abbr;
run;
quit;
%end;
%mend merge_neighbor;
%merge_neighbor;

```

After the merging is completed, the distance is calculated between each county centroid's latitude and longitude and the station's latitude and longitude. Distance is calculated using the Great Circle distance calculation method [3]. The macro "distance_calc" completed this calculation:

```

%macro distance_calc;
%do i=1999 %to 2005;
/*produce data set including distance from county centroid to each station*/
data v_&i(drop=lat1 lat2 long1 long2) v_ak_&i(drop=lat1 lat2 long1 long2);
set vt&i;
lat1=latdecimal1*constant('PI')/180;
lat2=county_lat*constant('PI')/180;
long1=londecimal1*constant('PI')/180;
long2=abs(county_long)*constant('PI')/180;
distance=1.150779*60*(180/CONSTANT('pi'))*acos(sin(lat1)*sin(lat2)+cos(lat1)*
cos(lat2)*cos(long1-long2));
if state='AK' then output v_ak_&i;
else if distance<=80 then output v_&i;
%end;
run;
%mend distance_calc;
%distance_calc;

```

In order to save storage space and SAS running time, only weather stations within a particular distance to county centroid were kept. The distance between each station to county centroid in Alaska is much larger than that of other states. Data including distance information from Alaska was outputted separately, and then the Alaska file was appended back to the main file by using macro "append_temp":

```

%macro append_temp;
%do i=1999 %to 2005;
proc append base=v_&i data=v_ak_&i;
%end;
run;
%mend append_temp;
%append_temp;

```

After calculating the distance, the data were sorted by state FIPS, county FIPS, and distance using macro "sort_temp":

```

%macro sort_temp;
%do i=1999 %to 2005;
proc sort data=v_&i;
by state_fips county_fips distance;
%end;
run;
%mend sort_temp;
%sort_temp;

```

By running macro "count_six," temperature or precipitation data from six stations nearest to each county centroid were kept to estimate county-level temperature and precipitation.

```

%macro count_six;

```

```

%do i=1999 %to 2005;
data temp.county_temp&i;
set v_&i;
by state_fips county_fips;
if first.county_fips then count=0;
count+1;
if count le 6 then output;
%end;
run;
%mend count_six;
%count_six;

```

Then average temperature from six nearest stations was calculated.

Taking 2005 temperature data as an example, by choosing six nearest stations, weather stations in 43 states and territories were located within 50 miles maximum distance to county centroid, and weather stations in 9 states and territories were located within 80 miles maximum distance to county centroid.

The macro "average2" and macro calls that follow would complete this calculation for 2000 and 2004 temperature data:

```

%macro average2(data=, set=);
data &data(keep=(keep=neighboring_state neighboring_state_abbr year state_fips
county county_fips pop county_lat county_long elev t_ave_jan1-t_ave_jan31
t_ave_feb1-t_ave_feb28 t_ave_mar1-t_ave_mar31 t_ave_apr1-t_ave_apr30 t_ave_may1-
t_ave_may31 t_ave_jun1-t_ave_jun30 t_ave_jul1-t_ave_jul31 t_ave_aug1-
t_ave_aug31 t_ave_sep1-t_ave_sep30 t_ave_oct1-t_ave_oct31 t_ave_nov1-t_ave_nov30
t_ave_dec1-t_ave_dec31);
array temp(365) jan1-jan31 feb1-feb28 mar1-mar31 apr1-apr30 may1-may31 jun1-jun30
jul1-jul31 aug1-aug31 sep1-sep30 oct1-oct31 nov1-nov30 dec1-dec31;
array t_ave_temp(365) t_ave_jan1-t_ave_jan31 t_ave_feb1-t_ave_feb28 t_ave_mar1-
t_ave_mar31 t_ave_apr1-t_ave_apr30 t_ave_may1-t_ave_may31 t_ave_jun1-t_ave_jun30
t_ave_jul1-t_ave_jul31 t_ave_aug1-t_ave_aug31
t_ave_sep1-t_ave_sep30 t_ave_oct1-t_ave_oct31 t_ave_nov1-t_ave_nov30 t_ave_dec1-
t_ave_dec31;
array s_temp(365) s1-s366;
array f_temp(366) f1-f366;
set &set;
by state_fips county_fips;
retain s1-s366 f1-f366
if first.county_fips then do;
do i=1 to 366;
t_ave_temp(i)=0;
s_temp(i)=0;
f_temp(i)=0;
end;
end;
do i=1 to 366;
if temp(i)=. then do;
s_temp(i)=s_temp(i);
f_temp(i)+0;
end;
else do;
s_temp(i)+temp(i);
f_temp(i)+1;
end;
end;
if last.county_fips then do;
do i=1 to 366;
if f_temp(i)=0 then t_ave_temp(i)=.;
else t_ave_temp(i)=s_temp(i)/f_temp(i);

```

```

        end;
        output;
    end;
    run;
%mend average2;
%average2(data=temp.temp2004ave6s, set=temp.county_temp2000);
%average2(data=temp.temp2004ave6s, set=temp.county_temp2004);

```

3) ESTIMATE MAXIMUM DAILY TEMPERATURE AND PRECIPITATION AT EACH COUNTY CENTROID USING ORDINARY KRIGING

Step 1: Set up temperature data at county level containing stations from neighboring county and neighboring states.

This step can be done with the macro code named “merge_neighbor,” “distance_calc,” and “append_temp,” which was already used for creating county-level estimation based on the six nearest stations.

Step 2: Separate temperature and precipitation data by state, and remove the duplicate weather stations.

Ordinary kriging could not be done with all of the U.S. data within our computing constraints, but could be done by state. Therefore, the data was separated by state. The following macro completes this step for 2005 temperature data:

```

%let clist=AK AL AR AZ CA CO CT DC DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN
MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA PR RI SC SD TN TX UT VA VT WA WI WV WY;

%macro separate(set=,);
%do j=1 %to 52;
data temp05_%scan(&clist,&j);
set &set;
by neighboring_state_abbr;
if neighboring_state_abbr="%scan(&clist,&j,' ')" and distance<=max_dist then output;
%end;
%mend separate;
%separate(set=v_2005);

```

In calculating the maximum daily temperature by simply averaging the six nearest stations, a single weather station may be used for multiple counties. Therefore, some weather stations were duplicated in the data file used for that calculation. In order to create estimates using kriging, the duplicate weather stations for each state have to be removed from the data set. The next macro removed the duplicate stations from the data sets:

```

/*sort data by stationid*/
%macro sort;
%do j=1 %to 52;
proc sort data=temp05_%scan(&clist,&j);
by stationid;
%end;
%mend sort;
%sort;

/*remove the duplicate stations*/
%macro remove;
%do j=1 %to 52;
data station05_%scan(&clist,&j);
set temp05_%scan(&clist,&j);
by stationid;
if last.stationid then output;
%end;
%mend remove;
%remove;

```

Step 3: Merge temperature and precipitation of each state with county centroid data.

To prepare the data sets to create estimates using kriging, the Census county file with population-based latitude and longitude was divided by state, and then combined with each corresponding state's temperature or precipitation file. The following macros will accomplish these two tasks:

```
%macro county;
%do k=1 %to 52;
data county_%scan(&clist,&k);
set county.county_pop_1;
by state;
format lat 5.1 lon 6.1;
lat=county_lat;
lon=-(county_long);
where state="%scan(&clist,&k, ' ' )";
%end;
%mend county;
%county;

%macro krig_data;
%do k=1 %to 52;
data virtual.krig05_%scan(&clist,&k);
set station05_%scan(&clist,&k) county_%scan(&clist,&k);
%end;
%mend krig_data;
%krig_data;
```

Step 4: Separate data by date.

We need to get a predicted value for each county's daily maximum temperature and precipitation by ordinary kriging. The set of stations with complete data varies for each date. Therefore, we separate the data into sets by date using the macro below:

```
%macro Krig_month(day, mon);
%do k=1 %to 52;
%do m=1 %to &mon;
%do i=1 %to &day;
data pred.fit_%scan(&clist,&k)%scan(&month,&m)&i(keep=stationid year
neighboring_state_abbr state_fips
county county_fips latdecimal1 londecimal1 lat lon county_lat county_long pop
%scan(&month,&m)&i);
set virtual.krig05_%scan(&clist,&k)(where=(%scan(&month,&m)&i ne .))
county_%scan(&clist,&k);
%end;
%end;
%end;
run;
%mend Krig_month;
%let month=jan mar may jul aug oct dec;
%Krig_month(31,7);
%let month=apr jun sep nov;
%Krig_month(30,4);
%let month=feb;
%Krig_month(28,1);
```

Step 5: Ordinary Kriging with Proc Mixed

After completing step 4, each date-specific data set contains the temperature of weather stations from a single state and its neighboring states, latitude and longitude information for each weather station and each county centroid;

however, the temperature for those county centroids is missing. The purpose of ordinary kriging is to obtain a predicted value for the daily maximum temperature for each county centroid for each date. Ordinary kriging can be done with SAS Proc Variogram, Proc Krige2d [4], or Proc Mixed [5].

With Proc Krig2d, some parameters such as nugget, sill and range of distance have to be calculated first by using Proc Variogram for each model. Proc Variogram involves manually observing plots of data and selecting the parameters. This method is useful for fitting a relatively small number of models, but it is not feasible for fitting many models needed for this project (one for each of 365 days in 7 years for each state). However, fitting spatial models with Proc Mixed does not require the parameters used by Proc Krige2d, and the procedure can be easily repeated by using SAS macro. Therefore, we used ordinary kriging with Proc Mixed to obtain county-level daily maximum temperature estimates for 50 states, DC, and Puerto Rico. Three commonly used covariance structures (spherical, exponential, and Gaussian) for spatial models are available from Proc Mixed. From the fit statistics of these models, we found that the exponential covariance structure best fits our data. The corresponding macro "mix_month" follows here:

```
%macro mix_month(day, mon);
%do k=1 %to 52;
%do m=1 %to &mon;
%do i=1 %to &day;
title "Fit the mixed model of %scan(&clist,&k)%scan(&month,&m)&i";
proc mixed data=pred.fit_%scan(&clist,&k)%scan(&month,&m)&i;
  model %scan(&month,&m)&i= / outp=pred.pred05_%scan(&clist,
&k)_%scan(&month,&m)&i;
  repeated / subject=intercept type=sp(exp)(lat lon);
  ods select Dimensions NObs FitStatistics
      ConvergenceStatus CovParms;
%end;
%end;
%end;
run;
%mend mix_month;

%let month=jan mar may jul aug oct dec;
%mix_month(31,7);
%let month=apr jun sep nov;
%mix_month(30,4);
%let month=feb;
%mix_month(28,1);
```

Step 6: Keep observations with predicted values; append data together by date and state.

The four macros listed next accomplish these tasks: keep observations with predicted temperature or precipitation value for each county centroid; append daily predicted temperature by state; and append the predicted data from all states together.

```
%macro keep_pred(day, mon);
%do k=1 %to 52;
%do m=1 %to &mon;
%do i=1 %to &day;
data fin.k05_%scan(&clist, &k)_%scan(&month,&m)&i(keep=state_abbr year state_fips
county_fips
county county_lat county_long lat lon %scan(&month,&m)&i std_%scan(&month,&m)&i);
set pred.pred05_%scan(&clist, &k)_%scan(&month,&m)&i;
format std_%scan(&month,&m)&i 5.2;
if %scan(&month,&m)&i=. then do;
state_abbr="%scan(&clist,&k,' ' )";
year=2005;
%scan(&month,&m)&i=pred;
std_%scan(&month,&m)&i=stderrpred;
output;
end;
```

```

%end; %end; %end; run;

%mend keep_pred;
%let month=jan;
%keep_pred(31,1);
/*... repeat for other months ...*/

%macro append_vars(day, mon);;
%do k=1 %to 52;
%do m=1 %to &mon;
%do i=1 %to &day;
data temp.pred05_%scan(&clist, &k);
merge temp.pred05_%scan(&clist, &k) fin.k05_%scan(&clist,
&k)_%scan(&month,&m)&i(keep=state_fips county_fips
%scan(&month,&m)&i std_%scan(&month,&m)&i );
by state_fips county_fips;
%end; %end; %end;
run;
%mend append_vars;
%let month=jan;
%append_vars(31,1);
/*... repeat for other months ...*/

%macro create_emp;
%do k=1 %to 52;
proc append base=temp.pred05_%scan(&clist, &k)(keep=state_abbr year state_fips
county_fips state_fips county_fips county county_lat county_long lat lon);
/*attrib feb29 format=3. informat=3. length=3;*/
set fin.k05_%scan(&clist, &k)_jan1;
/*feb29=.;*/
%end;
run;
%mend create_emp;
%create_emp;

data temp.pred05;
set temp.pred05_ak(obs=0);
run;
%macro append_state;
%do k=1 %to 52;
proc append base=temp.pred05 data=temp.pred05_%scan(&clist, &k);
%end;
run;
%mend append_state;
%append_state;

```

After these macros were run, county-level estimates of daily maximum temperature are stored in one file for year 2005.

4) COMPARE ESTIMATES FROM SIMPLE AVERAGE TO ORDINARY KRIGING

Step 1: Merge average estimated temperature file with kriging estimated temperature file.

Through step 2) and step 3), we obtained two files containing estimates for maximum daily temperature by the simple average and ordinary kriging methods, respectively. Sorting these two data sets by state and county FIPS codes, then merging by these variables, we create a combined file with estimates for daily maximum temperature by average and kriging methods for 3,219 counties and county equivalents. The SAS code for this step is simple and so was omitted.

Step 2: Measure the correlation between average estimated temperature and kriging estimated temperature.

The SAS macro "reg_pred_ave" uses Proc Reg with a plot option to measure correlation between county-level estimates for temperature by the average and kriging methods. ANOVA table, root Mean Square Error(MSE), adjusted R Square(Adj-R-Sq), and a regression plot of the average estimates versus the kriging estimates were included in the SAS output.

```

SYMBOL1 V=circle C=blue I=r;
%macro reg_pred_ave(day, mon);
%do m=1 %to &mon;
ods rtf file="C:\noaa\temperature data\regplot2005%scan(&month,&m).rtf";
%do i=1 %to &day;
TITLE1 "Predicted vs Simple Average on %scan(&month,&m)&i";
proc reg data=temp.pred_ave_2005;
model %scan(&month, &m)&i=ave_%scan(&month,&m)&i;
plot %scan(&month, &m)&i*ave_%scan(&month,&m)&i;
run;
%end; %end;
run;
quit;
ods rtf close;
%mend reg_pred_ave;
%let month=jan;
%reg_pred_ave(31,1);
...
%let month=dec;
%reg_pred_ave(31,1);

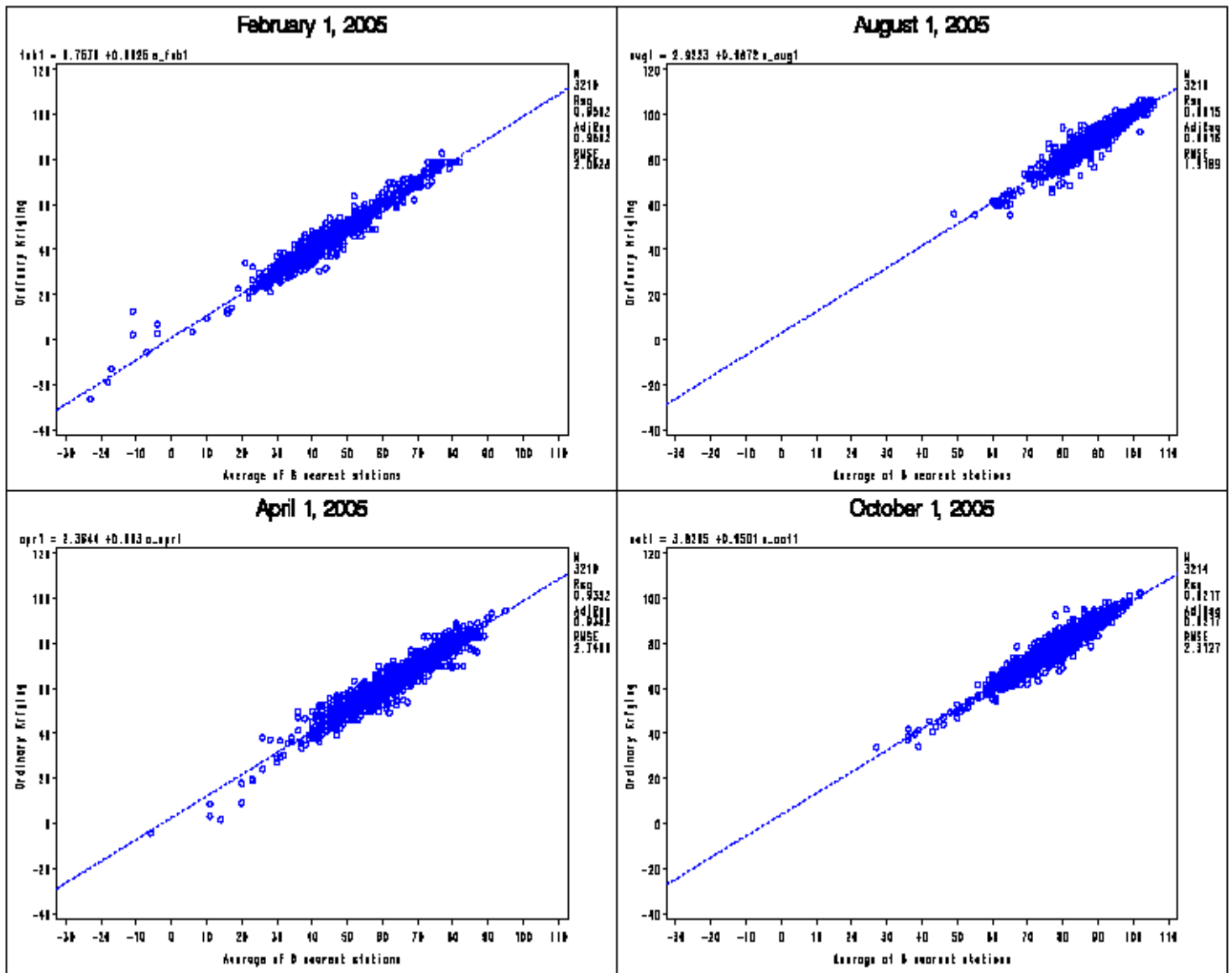
```

Kriging estimates (predicted) were plotted against average estimates for the maximum daily temperatures for February 1, April 1, August 1, and October 1 of 2005 in the following examples. The adjusted R square (AdjRsqr) for these 4 days are all larger than 0.9, indicating a strong correlation between county-level estimates from these two methods.

CONCLUSION

This paper demonstrated procedures and processes used to create county-level estimates for maximum daily temperature and precipitation from National Weather Service data. Many SAS procedures were involved and macros were developed to accomplish the task. Ordinary kriging with SAS Proc Mixed made the estimation more efficient as compared to SAS KRIG2D. Strong correlation between average estimated and kriging estimated temperature indicated that these methods produce similar results. We hope this presentation will prove useful for analysts working with National Weather Service data in conjunction with county-level data from other sources, or other large collections of data stored in many files.

Figure. Plots of simple average and ordinary kriging estimates for maximum daily temperature at centroids of counties and county equivalents in the United States for selected dates in 2005



REFERENCES

1. NCDC, 2003: Data documentation for data set 3200 (DSI-3200): Surface land daily cooperative summary of the day. National Climatic Data Center, Asheville, NC, 36 pp. [Available online at <http://www.ncdc.noaa.gov/pub/data/documentlibrary/tddoc/td3200.pdf>.]
2. U.S. Census Bureau. Census 2000 U.S. Gazetteer Files. [Available online at <http://www.census.gov/geo/www/gazetteer/places2k.html>].
3. The North American Association of Central Cancer Registries (NAACCR). Great Circle Distance Calculator SAS code <http://www.naacr.org/filesystem/word/GreatCircleDistances4.doc>
4. SAS Institute Inc. (2005) SAS OnlineDoc 9.1.3. Cary, NC: SAS Institute Inc.

5. Littell RC, Milliken GA. SAS for Mixed Models. 2nd Ed. Cary, NC: SAS Institute Inc., 2006

ACKNOWLEDGMENTS

This work was supported by the Dental, Oral and Craniofacial Data Resource Center, a joint project of the Center for Disease Control and Prevention's (CDC's) Division of Oral Health and the National Institute of Health's (NIH's) National Institute of Dental and Craniofacial Research. The National Oceanic and Atmospheric Administration's National Climatic Data Center provided the National Weather Service data. The authors wish to acknowledge the generous guidance of Carol Gotway Crawford, James Holt and Robert Schwartz from CDC on selection of statistical methods, geographic data, and computing resources.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Liang Wei

The Ginn Group/Northrop Grumman

In support of the Dental, Oral, and Craniofacial Data Resource Center

A joint project of the Division of Oral Health, National Center for Chronic Disease Prevention and Health Promotion, Centers for Disease Control and Prevention, and the National Institute of Dental and Craniofacial Research, National Institutes of Health

Work Phone: (770) 488-6070

E-mail: Gjd5@cdc.gov

Or

Laurie Barker

Surveillance, Investigations, and Research Team

National Center for Chronic Disease Prevention and Health Promotion

Coordinating Center for Health Promotion

Centers for Disease Control and Prevention

Work Phone: (770) 488-5961

E-mail: LBarker@cdc.gov

4770 Buford Hwy, MS F-10

Atlanta, GA 30341

Fax: 770-488-6080

Web: drc.hhs.gov

SAS and all other SAS Institute, Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.