AWR Contract No. AB-133E-16-CQ-0024 AWR2

Development of Census Tract Level Daily Climate Data

Sept. 1, 2020 – May 1, 2021

Principal Investigator

Karsten Shein, PhD
Director, Midwestern Regional Climate Center
University of Illinois
Illinois State Water Survey
2204 Griffith Drive
Champaign, IL 61820
Tel: (217) 244-1151
Email: kshein@illinois.edu

Coauthors

C. Travis Ashby
Senior Scientific Specialist
Midwestern Regional Climate Center
Illinois State Water Survey
University of Illinois

Zoe Zaloudek
Geospatial Application Developer
Illinois State Water Survey
University of Illinois

Mike Timlin
Regional Climatologist
Midwestern Regional Climate Center
University of Illinois

Abstract

We rescale nClimGrid-d temperature data to census tracts using a combination of commercially available GIS software and Python for computations and data management. Our approach utilizes areal averaging in which each census tract is comprised of nClimGrid-d grid (originating grid) cells. These sub-areas contribute not only temperature data but, through their area, an averaging weight. We discuss the advantages and limitations of this approach. An average rescaling error is computed by mapping the census tract estimates back to the originating grid and comparing to the original temperature values on the originating grid over specific time periods. The largest errors occur over the western United States, especially near complex terrain. The smallest errors are present over the Great Plains and Midwest U.S. (typically less than 1°C) Finally, the supplied computational code may be modified to compute different weighting methodologies based on other quantifiable factors (population, proximity to meteorologically significant bodies of water, etc..)

Originating Grid (nClimGrid-d)

nClimGrid-d is a daily, gridded dataset of precipitation and minimum, maximum and average temperature. The grid covers the contiguous United States for the period 1951 to 2021 and is a regularly spaced, latitude-longitude grid with spacing of 1/24 ° (0.041667 °). This yields a longitudinal grid spacing from 3.0 km on the northern boundary to 4.2 km along the southern boundary. Additional details on the background of the product may be found in the Product Description (Durre, 2018, ftp.ncei.noaa.gov:/pub/data/daily-grids/docs/nclimdiv-description.pdf.). Although it is suggested that the daily dataset has recently been extended back to 1895, we were unable to locate these additional data for inclusion in this work.

In this study, each grid point is taken to represent the temperature over the accompanying grid cell area. It should be noted here that an alternative, though more computationally expensive and memory intensive process could be used. Namely, this original dataset could be immediately scaled to a finer grid using an appropriate interpolation scheme (e.g. bilinear interpolation). If the finer grid cell areas are substantially smaller than the smaller census tracts, grid cell boundary artifacts will be reduced in the estimate.

Methods

Here we describe the two steps used in producing the temperature estimates.

Subdivision of domain into sub-areas

There are two basic spaces in this problem. The first is the census tract (CT) space and the second is the gridded, or grid point (GP) space. Each element or tract in CT space is uniquely identified by the Geographic Identifier (GeoID) as described by the United States Census Bureau: (https://www.census.gov/programs-surveys/geography/guidance/geo-identifiers.html and the Federal Communications Commision:

https://transition.fcc.gov/form477/Geo/more_about_census_tracts.pdf

In short, the 11-digit code consists of a 2-digit state code followed by a 3-digit county code followed by a 6-digit tract code. The first 4 digits of the tract code and the last 2 digits are separated by

an implied decimal. Hence census tract 106.01 in Champaign county (county code = 19), Illinois (state code = 17) is designated by the GEOID 17019010601. Likewise, the unique identifier for a GP element is the ordered pair (i, j), $0 \le i < 1385$, $0 \le j < 596$. The GIS component of the subdivision process is described below.

GIS (ArcGIS) Component of Sub-area Identification

See script "1_DownloadAndAggregate_CensusTracts.py":

This script downloads 2020 Census Tract data from the US Census by state (for the Lower 48 states). The data is stored online in zipfiles available at https://www2.census.gov/geo/tiger/TIGER2020/TRACT. Each shapefile is extracted from each downloaded zipfile. A file geodatabase (named USCensusTracts2020.gdb) and an empty polygon feature class (CensusTracts2020) are created. The spatial reference used for the feature class (NAD 83 GCS) is copied from the first shapefile that was downloaded. The data from the individual state shapefiles is appended to the feature class. Last, a new field is added (AREATOTAL). Values for this field are calculated as the sum of the ALAND and AWATER fields (side note, these two fields are calculated by the US Census and are in square meters). The result is one polygon feature class with all 2020 US Census Tracts and their attributes. Note – the unique ID (UID) field for Census Tracts is named GEOID. Further info can be found in the Technical Documentation from the US Census Bureau (https://www2.census.gov/geo/pdfs/maps-data/data/tiger/tgrshp2020/TGRSHP2020_TechDoc.pdf).

See script "2_CreateGISPointsFromPNTfiles.py":

This script creates a file geodatabase (named nClimGridPoints.gdb) and an empty point feature class (nClimGridPoints). The spatial reference used for the feature class is the WGS 84 GCS. This spatial reference was chosen because NetCDF files of similar data downloaded from NCEI (ftp://ftp.ncdc.noaa.gov/pub/data/daily-grids/) were using this geographic coordinate system. Next, the script goes through each line in the provided pnt file named "201312.tave.conus.pnt". For each line in the file, the script strips out the latitude and longitude values and adds a point at that location to the feature class. A unique ID (UID) is also calculated using the lat/lon values. Next, the row and column (i/j) for each point is estimated, using the minimum lat/lon values found in the dataset and a presumed cell size of 0.0416667. A UID is also calculated using the i/j estimates. The result is a point feature class with all points found in the above-named pnt file and UID attributes.

See script "2p5_CreateTestData.py":

This script uses the Select tool to create a subset of the US Census Tracts and Grid Points for testing. Clauses for Champaign County IL and the State of Illinois have been set up in the script. The subsets are created as separate feature classes, located in the same geodatabase as the original, with "_forTEST" added to the new feature class name. This script does not need to be run when the user wants to run Step 3 for all of CONUS.

See script "3_CreateOutputFiles.py" located in:

This script runs geoprocessing tools and creates output CSV files. The projection chosen to work in is "USA Contiguous Albers Equal Area Conic USGS version". This was chosen because it matches the project area, has a linear unit of meters, area distortion is minimized (within the standard parallels of 29.5° and 44.5°), and is a relatively well-known projection. A file geodatabase (named GISprocessing.gdb) for working is created. If the US Census Tracts are not already in the above-described projection, it is reprojected using the Project tool and put in the working geodatabase. Next, the maximum and minimum lat/lon values of the Grid Points are found and used with the cell size value to set the spatial extent environment setting. A raster dataset is created from the grid points using the Point to Raster tool, with the unique IDs of the points assigned to the cell values. The raster is then converted to a polygon feature class using the Raster to Polygon tool, which is then also reprojected to the working projection. Now we have a set of tract polygons and a set of grid cell polygons in the same projected coordinate system.

There are multiple geoprocessing tools in ESRI's Analysis Toolbox Overlay Toolset that can be used to find overlap between polygons. The tool chosen for this project is Union. This tool will keep all areas from both input polygons, regardless of whether they cover each other. Other geoprocessing tool options include Identity and Intersect (see example figures at bottom of this section).

Finally, the script creates three CSV files. The first (Tracts.csv) is a list of the reprojected Census Tracts with their unique IDs, Census-calculated areas, and GIS-derived areas (both area values are in square meters). The second (GridCells.csv) is a list of the reprojected grid cell polygons with UIDs and GIS-derived areas (also in square meters). The third CSV (Combinations_Union.csv) lists every polygon in the results from the Union tool. Attributes included are the UIDs for tract and grid cell, GIS-derived areas (in square meters), and lat/lon values from the applicable grid points. All values in all CSVs are padded with spaces on the left as needed for 18-character width fields.

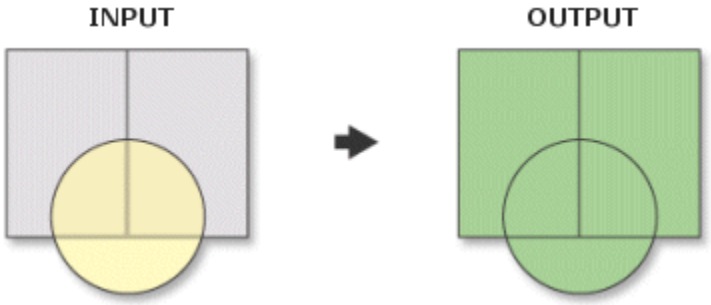

Figure 1.– Input/Output from Union Tool: https://desktop.arcgis.com/en/arcmap/10.6/tools/analysis-toolbox/GUID-6C93B42C-3D0B-4A7F-A9C7-4053D146CCB6-web.gif

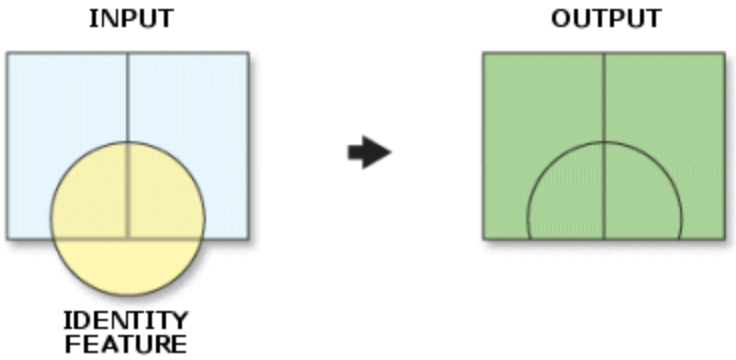

Figure 2. – Input/Output from Identity Tool: https://desktop.arcgis.com/en/arcmap/10.6/tools/analysis-toolbox/GUID-1721C45E-E0F9-441A-A1EA-F0949504CCAA-web.gif

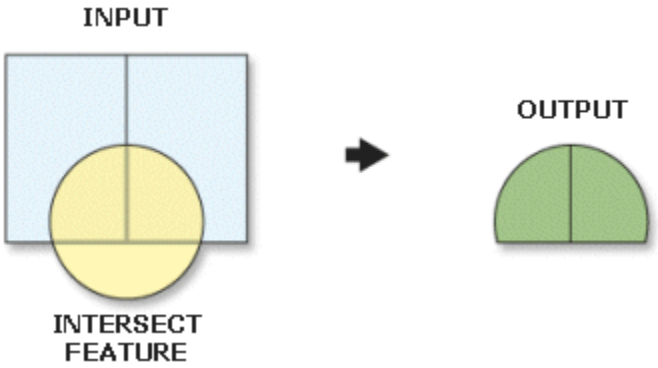

Figure 3– Input/Output from Intersect Tool: https://desktop.arcgis.com/en/arcmap/10.6/tools/analysis-toolbox/GUID-90AA6079-D9FD-41B6-AA87-F660C2FBB4AD-web.gif

At this point we have the subareas in which each subarea:

a.) Belongs to a particular GP element AND one CT element, or

b.) Belongs to a particular GP element but belongs to no CT element, or

c.) Belongs to a CT element, but belongs to no GP element

Figure 4 shows an example of the GIS located 2020 census tracts, gridpoints and grid cells.

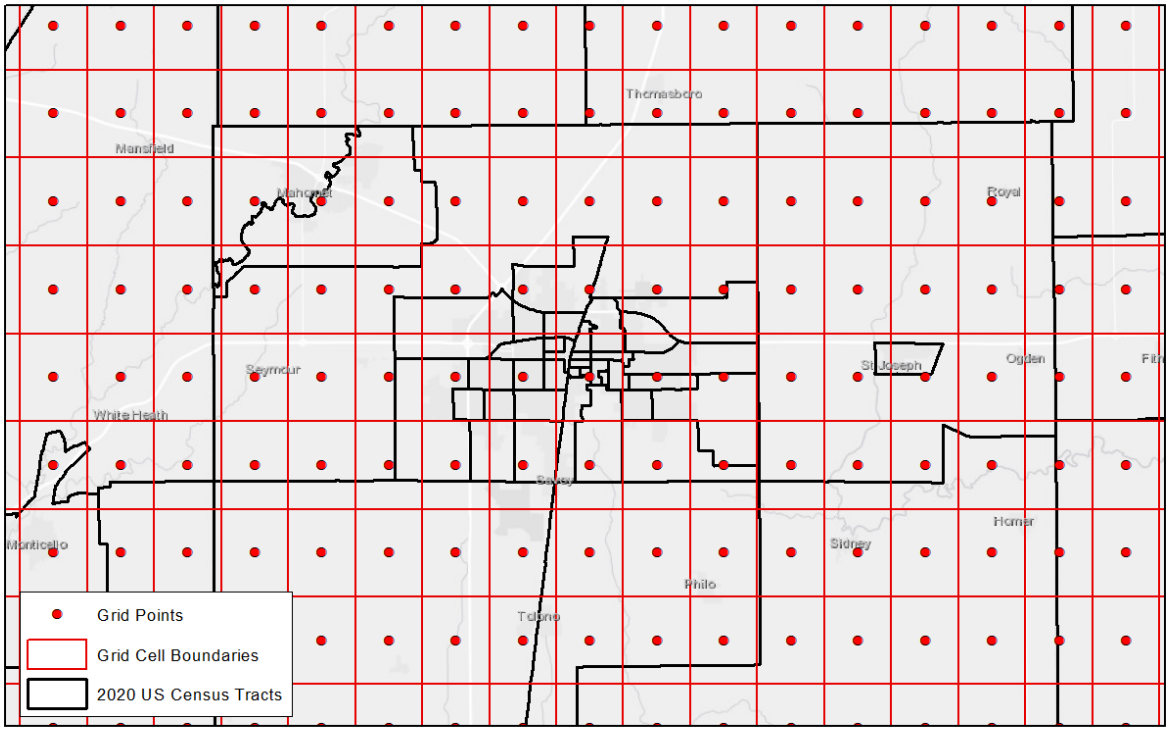

Figure 4. – Example of census tracts, grid points and grid cells located by GIS software. Location is Champaign-Urbana and vicinity.

Creation of the temperature estimates

For the purpose of creating a single temperature estimate for one census tract we must first establish mappings and associated attributes between the CT and GP elements. This is accomplished with "mapproc.g2c.py". This script takes as input the file, "Combinations_Union.csv" generated by the GIS scripts described above and creates a dictionary with census GEOID's as the key. This dictionary entry has the form of a list with the following elements:

mdict[GEOID][0], Scratch element (float or None) serving various purposes

mdict[GEOID][1], List with variable length to store temperature data for one month (e.g. tmin)

mdict[GEOID][2], " (e.g. tmax)

mdict[GEOID][3], " (e.g. tavg)

mdict[GEOID][4: len(mdict[key]], each element is a list, each list corresponds to a GP element that intersects the CT specified by the dictionary key. So for example

mdict[key][4]=[(i,j), subarea], where the (i, j) is the GP cell identifier (type integer indices into grid array), and subarea is the numerical value (type=float) of the area of intersection (e.g. in square meters). This would be the first contributing GP. This continues as mdict[GEOID][5], . . . etc. for each GP that contributes to the CT.

With this g2c dictionary, any census tract that has intersecting GP cells will appear in the dictionary along with the identifier for the GP cell and the area it is contributing to the intersection. This dictionary is stored on disk as a pickled object and is modified when computing the estimates but only for the purpose of storing one month of data in memory while processing. The original pickle file is not intended to be altered as it contains the needed mapping information for producing estimates for all months.

The temperature estimates are produced by "forproc.py" (forward processing), which traverses the g2c dictionary. For each CT key, an estimate is produced by creating a weighted temperature T_{est}:

$$T_{est} = \frac{\sum_n T_n A_n}{\sum_n A_n},$$

where A_n is the area of the nth contributing GP area (i.e. mdict[key][n≥4][1]), T_n is the contributing GP temperature, and T_{est} is the estimate CT temperature. Note that the normalizing factor is NOT the total area of the census tract but the sum of *contributing areas*. In most cases the total area of the CT and sum of GP contributing areas are the same, but in the case where a CT is not completely covered by valid GP cell areas, an estimate is still produced that is at least unbiased (with respect to the input data) when T_n is constant or the averaging operator is independent of n. Weighting functions different than the areal average could also be used. For example, population density could multiply the subareas, if available, for a population weighted average.

Out of over 80,000 census tracts processed, there are 501 census tracts that have no contributing GP elements with defined temperature values. These are identified in separate output files with filenames containing the word "umatch" and correspond to CT's that lie on or within water boundaries. The final CT estimate output files do not contain these tracts, but only contain CT's with at least one contributing GP area with a defined temperature for all days in the month of that file. These files are in subdirectories identified by years and are of the form "fld-YYYYMM-census.txt", where "fld" is the field of interest. Aside from the one line header at the start of the file, the remaining lines are in the form of a year, month and day followed by all census tract estimates for that day (one per line), followed by another line for year, month and day+1, temperature data, etc.. The output produced starts on 1 January 1951 and continues to 31 March 2021. Once the nClimgrid-d data is available for 1895-1950 (they were not present on ftpprd at the time of this AWR), this code can be applied to these years by simply creating the appropriate directories and modifying a few constants at the top of the source file (see code comments).

As an example, the rescaled maximum temperature (tmax) is shown below for 1 June 2000, utilizing the 2020 Census Tracts (Fig. 5). In the western U.S., the effects of both highly variable terrain and relatively large census tracts can be seen in the estimate.

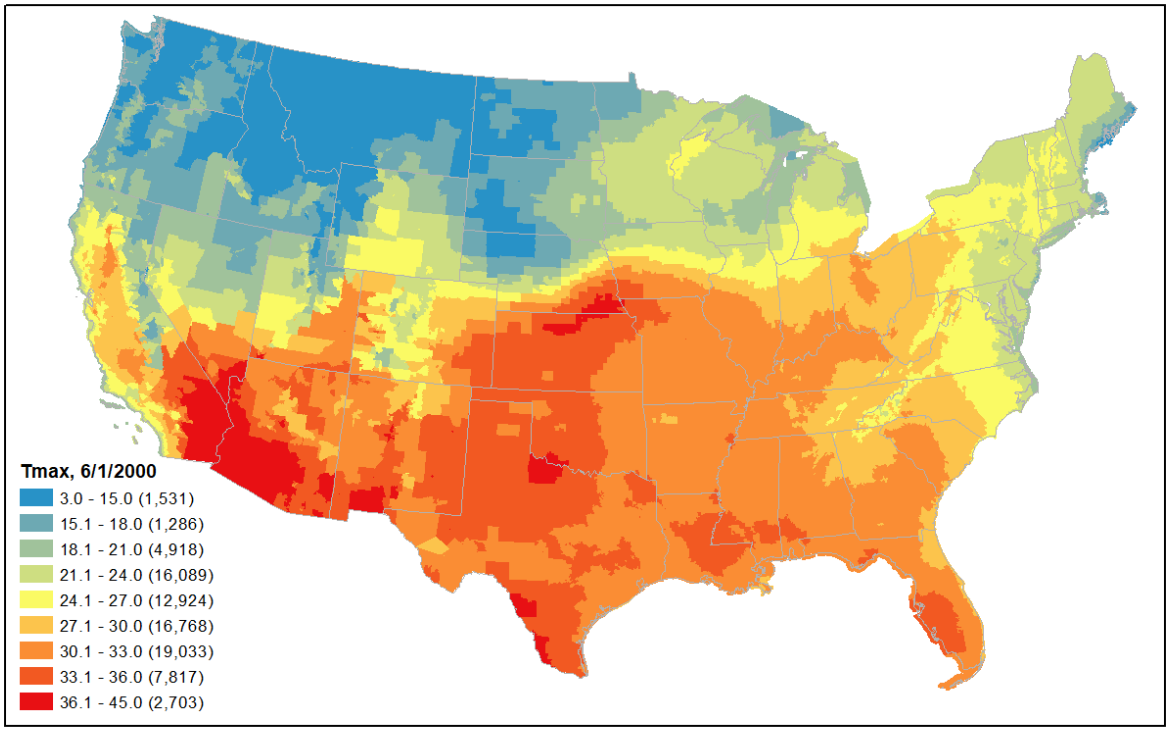

Figure 5. – Estimated maximum daily temperature (°C) for 01 June 2000 rescaled to 2020 census tracts from nClimgrid-d data. Census tract counts within each temperature bin are parenthesized.

Error Computation

The basic methodology used for error computation is based on the reverse mapping. That is, the entire GP set is estimated using the CT estimates with the same basic averaging procedure. For this purpose, the script "mapproc.c2g.py" creates a pickled dictionary keyed by an (i,j) tuple, in which each dictionary entry is a list. As in the g2c case, elements 4 onward belong to the CT elements that contribute area to the current GP element. Hence, cdict[(i,j)][4] = [GEOID, subarea], where GEOID is the unique GEOID (integer) of the contributing CT, and subarea is the area (float) that CT contributes to GP.

With this reverse mapping one can create a gridded estimate from the CT estimates generated earlier, again using areal averaging. Subtracting the original input grid from this estimate grid produces a gridded deviation for each day as specified at the beginning of the source file. These deviations are computed in the source file "devcomp.py" and are output as numpy arrays with the first two dimensions being the same as the nClimgrid-d originating grid, and day of month being the third dimension. Next the script, "errstat.py" will utilize the output files from the previous step to compute mean deviation, mean square deviation and root mean square (rms) deviation as a gridded output. Averaging periods are

specified at the beginning of the source file. Average deviation and rms deviation are shown in Figs. 6 and 7, where the period of averaging is all days in July for the 30-year period 1981-2010.

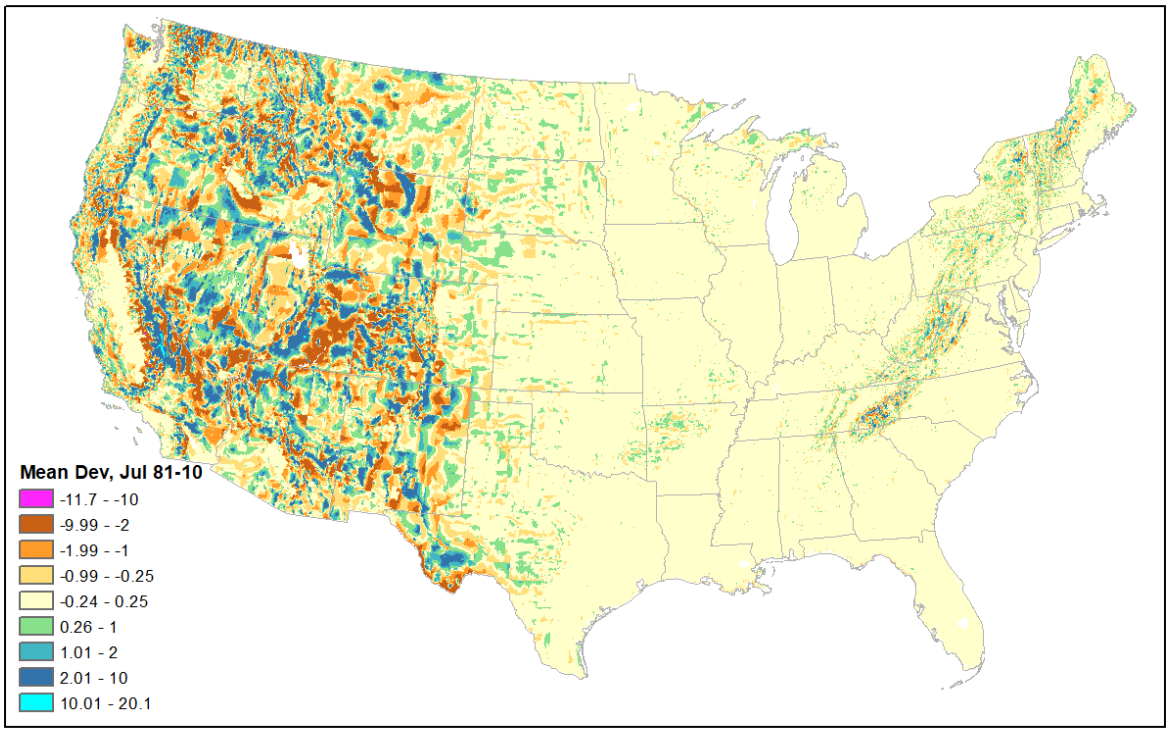

Fig. 6 – Mean deviation (°C) of tavg for the month of July, 1981-2010

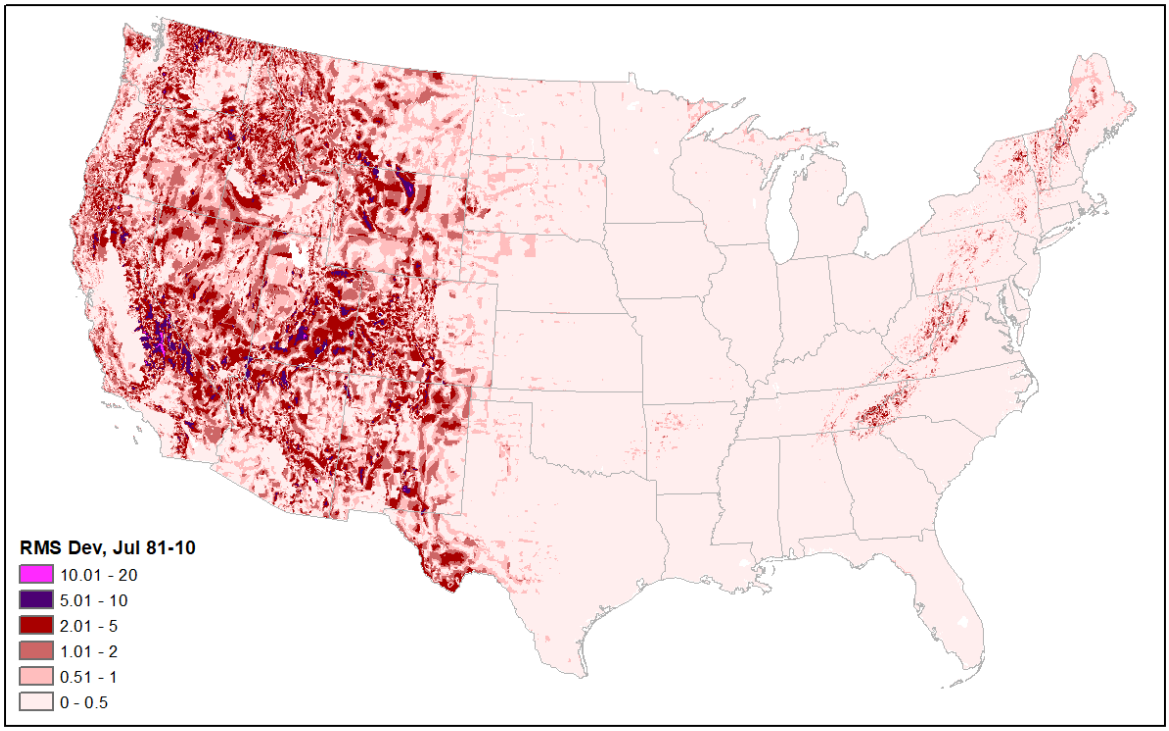

Fig. 7 – RMS deviation of average temperature, or tavg (°C) for July, 1981-2010.

As noted earlier, the mean and RMS deviations are largest over the western U.S. where terrain is highly variable and census tracts are larger. Over the Great Plains and portions of the Midwestern U.S., both statistics are much smaller and are often a fraction of a degree Celsius. It is paramount to understand that these deviations are not characterizing the CT estimate error relative to actual temperature, but are instead quantifying the inherent error associated with the estimation algorithm in going from 5-km gridded data to census tracts that may contain a large elevation range, have a large spatial extent and may be irregularly shaped. Before using this data for any application, one should examine the error expected from the gridded error statistics both in terms of bias relative to the input GP data and RMS error.

Conclusions

A methodology is implemented for estimating census tract temperatures (minimum, maximum, average) from the nClimgrid-d gridded datasets for the years 1951 – 2021. These estimates are precomputed and available, as is the code used. Error statistics (mean deviation, rms deviation) may be computed with the supplied code and examples are given in this report. Implicit interpolation errors are smallest over areas of the Great Plains and Midwest where elevation changes and census tracts are reduced relative to the larger errors of the Western United States.

The code supplied is meant to be modified as the user sees fit for their own application including using different weighting methods such as population weighting given an appropriate input dataset (e.g. gridded population density). Additional instructions explaining how to run code are included in the source files. Source files are available at https://github.com/isws-mrcc/nclim2census.git

Acknowledgments

The work described herein was supported in full by an Additional Work Request (AWR) Contract No. AB-133E-16-CQ-0024 AWR2 issued by the NOAA National Centers for Environmental Information (NCEI). This AWR is part of the NOAA Contract for the Midwestern Regional Climate Center (MRCC).