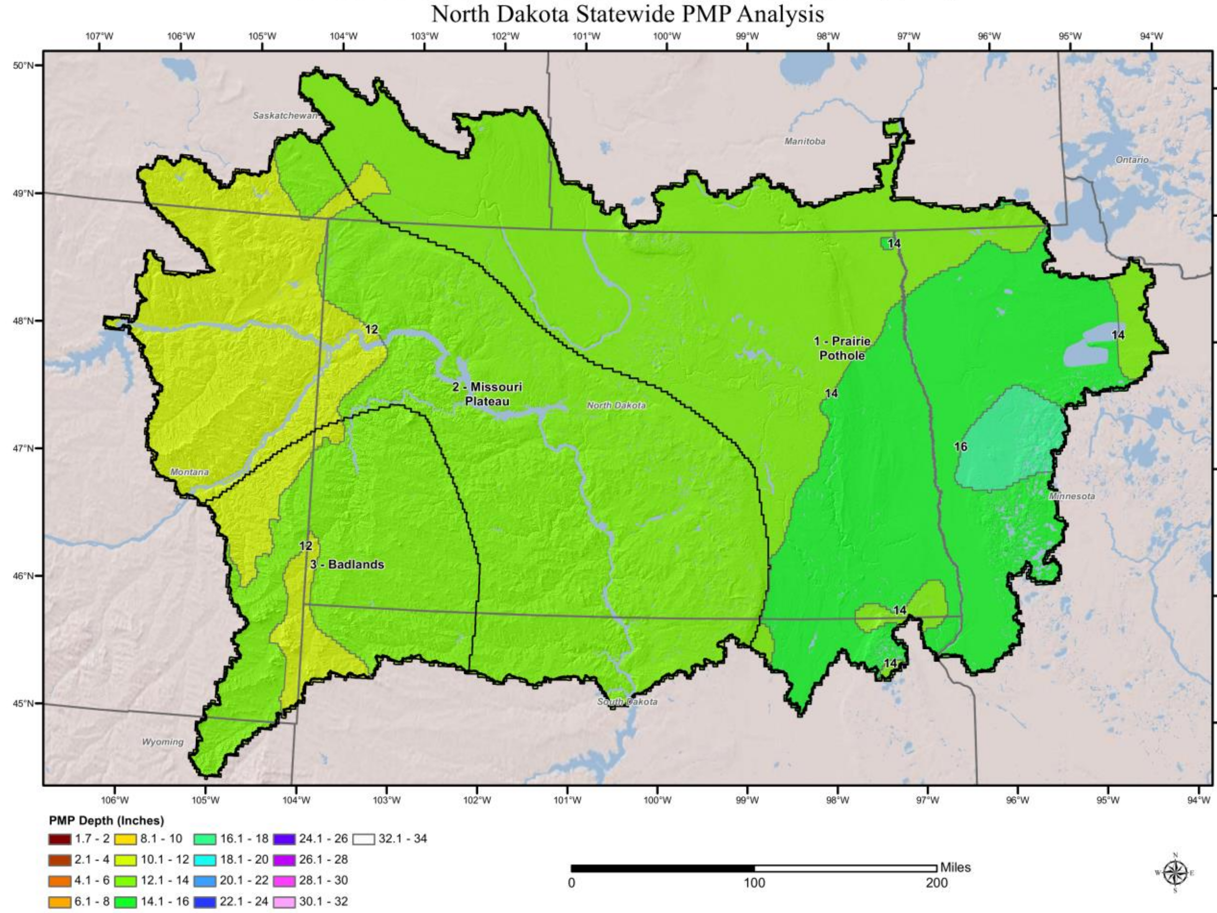
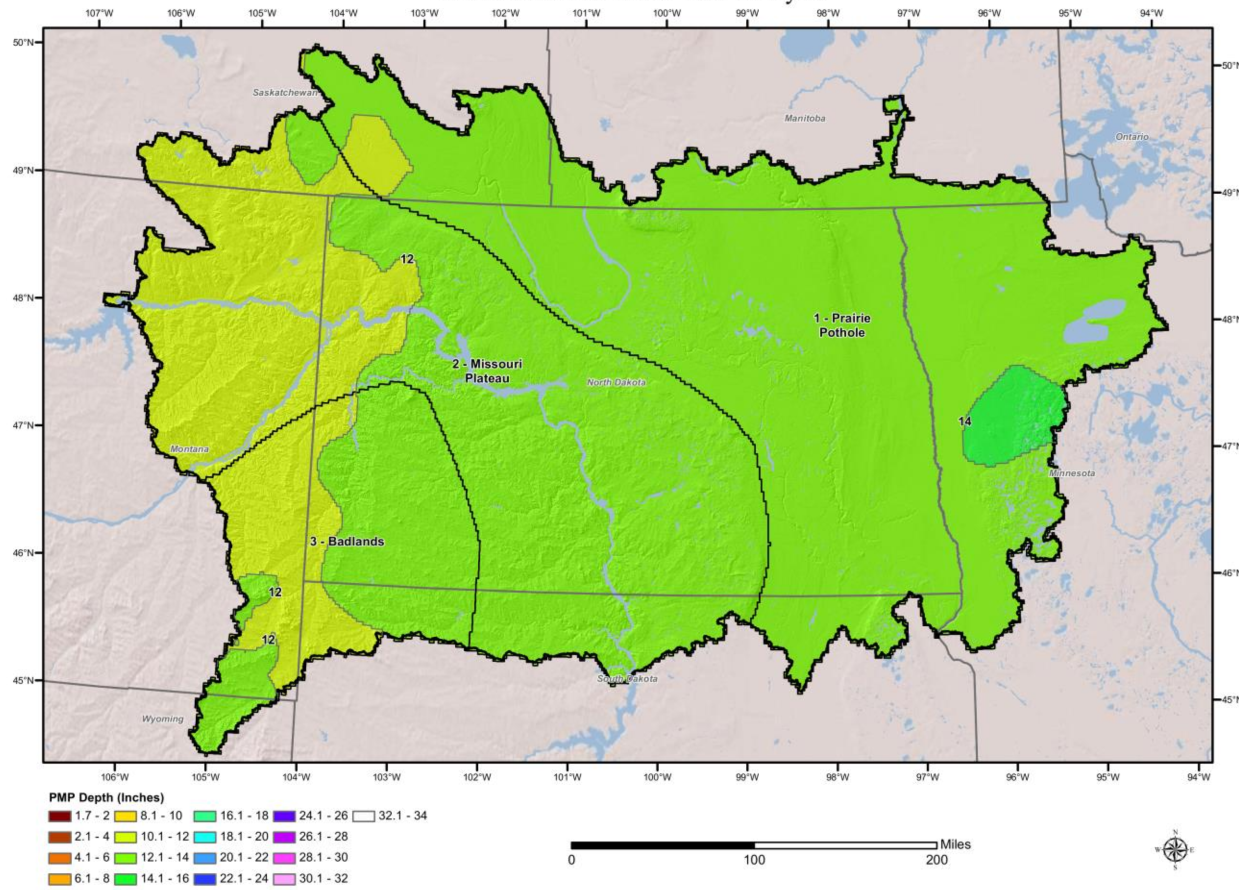
Appendix A
Probable Maximum Precipitation (PMP) Maps

General Storms

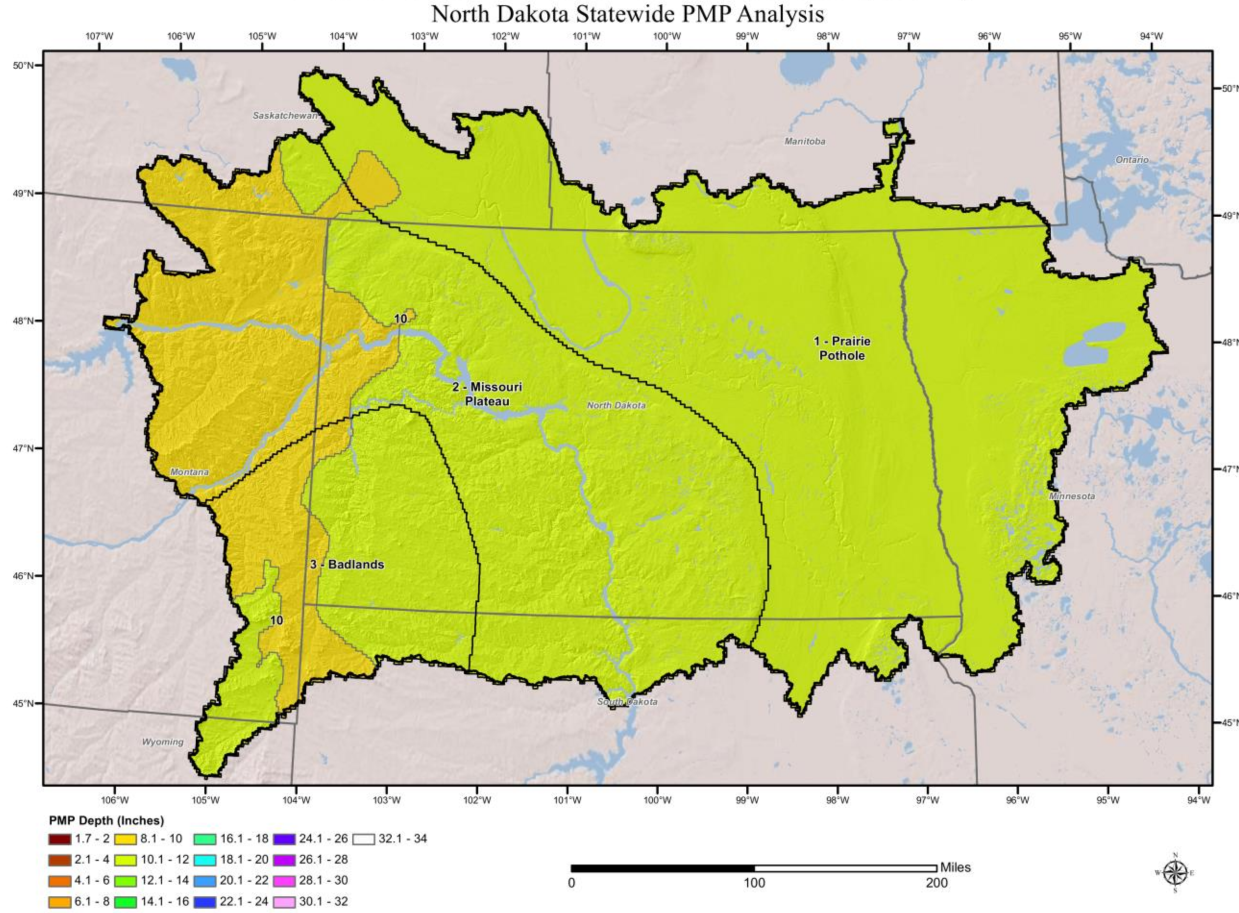
6-Hour General Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis

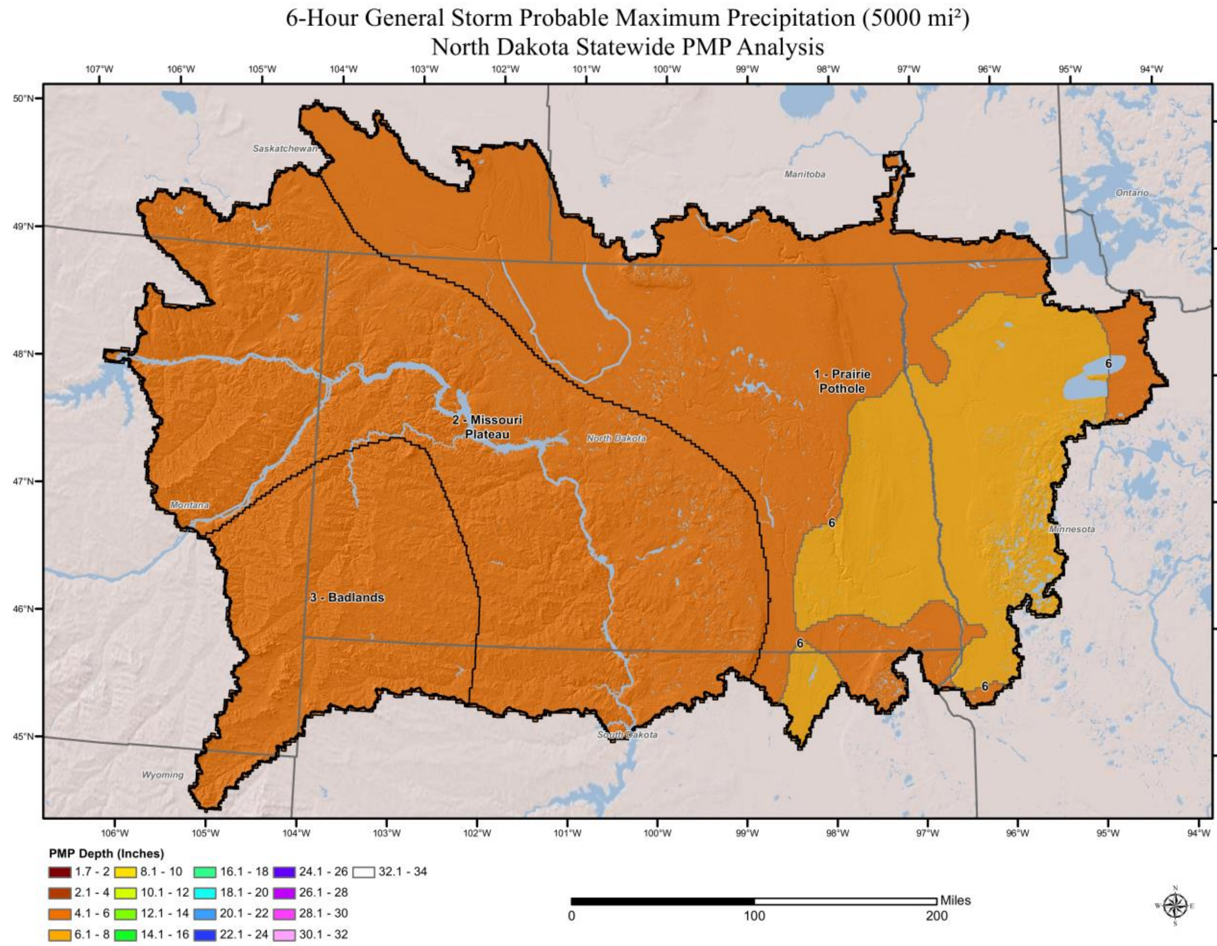


6-Hour General Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis

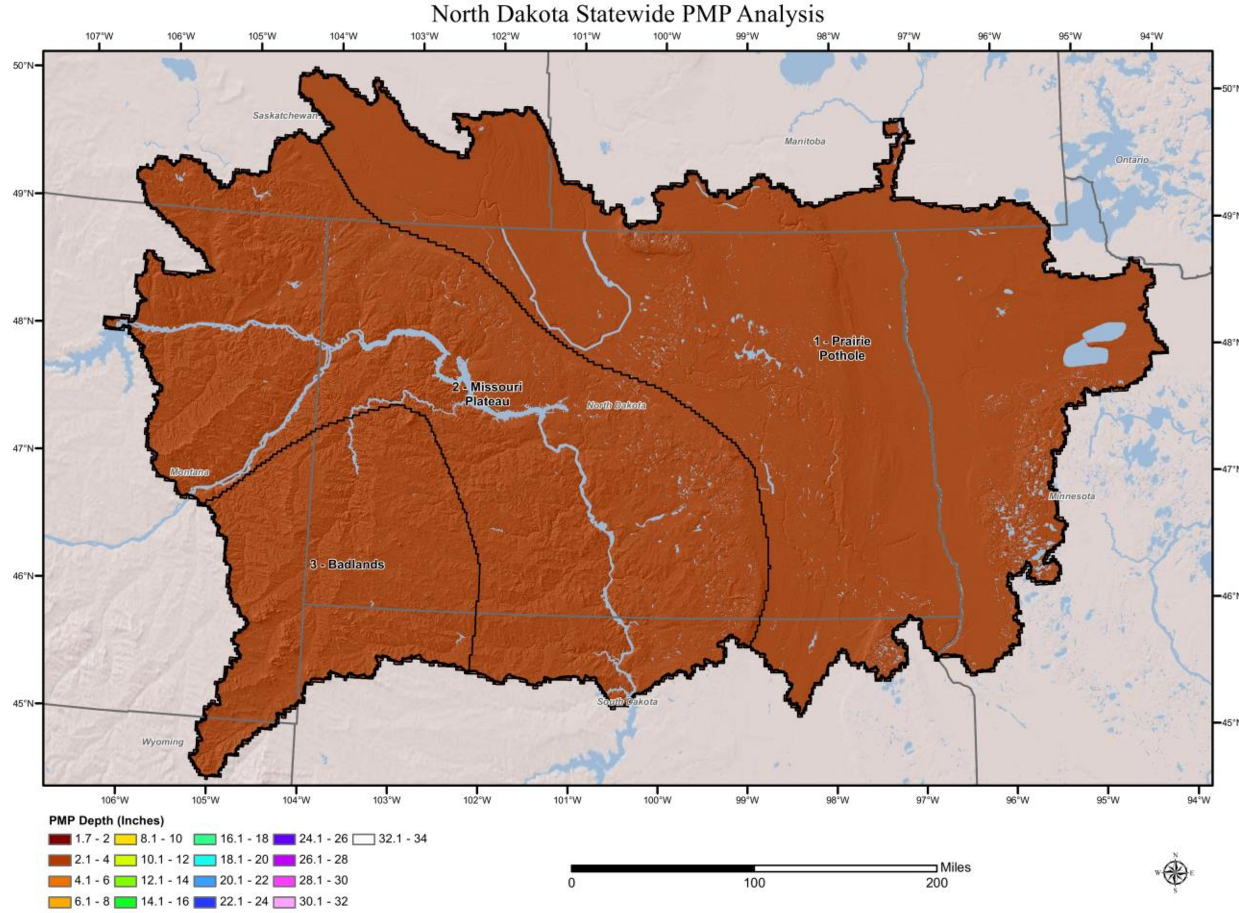


6-Hour General Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis

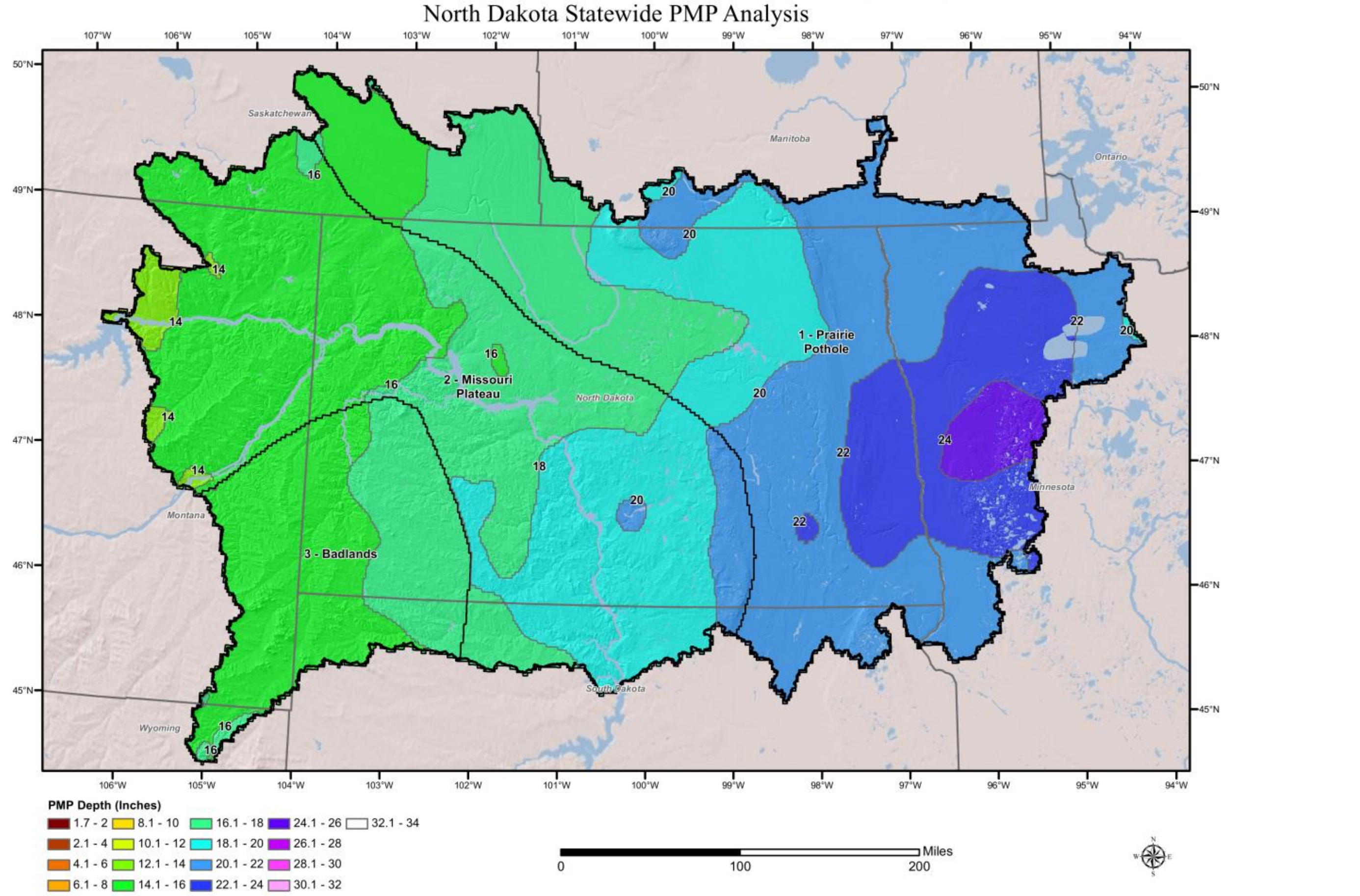



6-Hour General Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis
1 - Prairie Pothole
2 - Missouri Plateau
3 - Badlands
Saskatchewan
Manitoba
Ontario
Montana
North Dakota
Minnesota
South Dakota
Wyoming
PMP Depth (Inches)
1.7 - 2
2.1 - 4
4.1 - 6
6.1 - 8
8.1 - 10
10.1 - 12
12.1 - 14
14.1 - 16
16.1 - 18
18.1 - 20
20.1 - 22
22.1 - 24
24.1 - 26
26.1 - 28
28.1 - 30
30.1 - 32
32.1 - 34
Miles
0
100
200

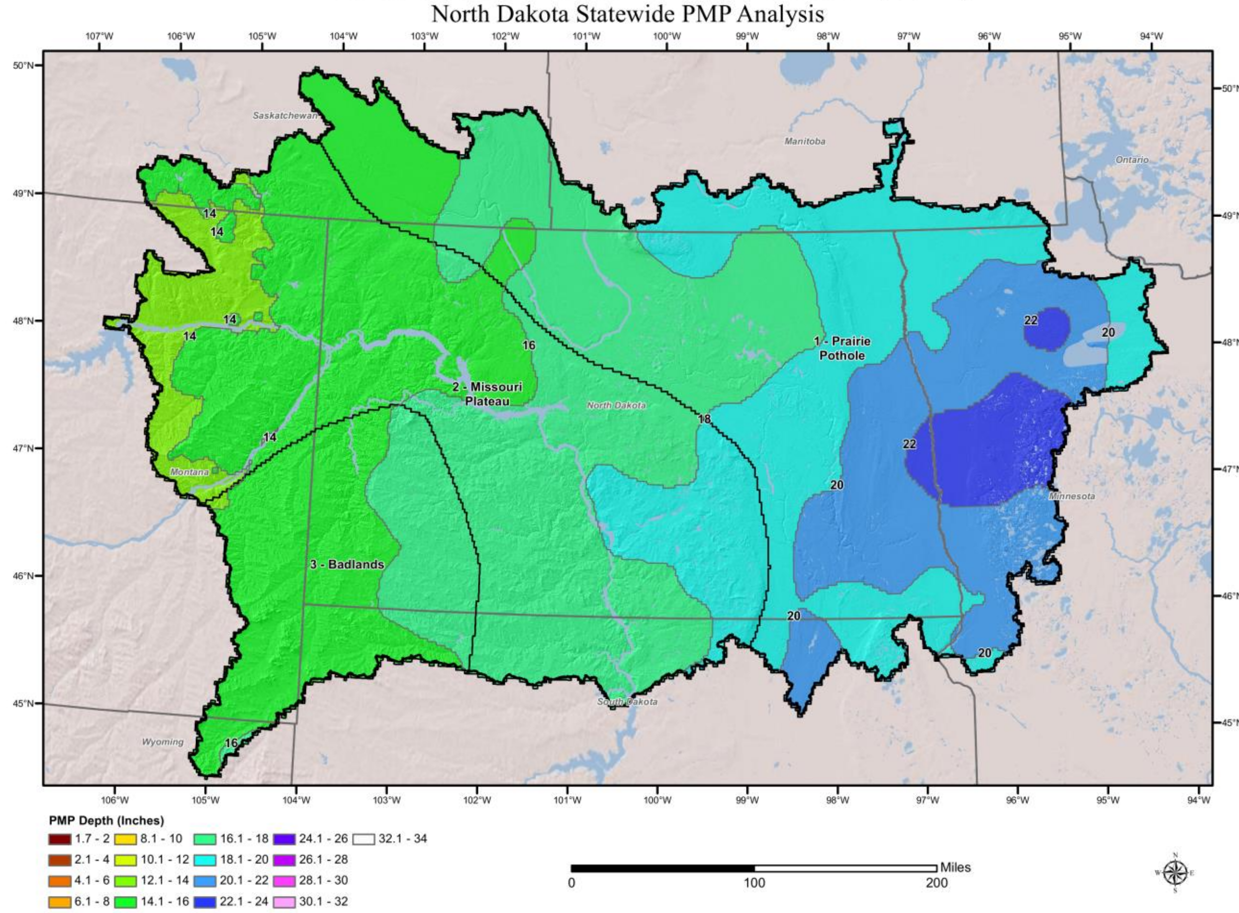
6-Hour General Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis



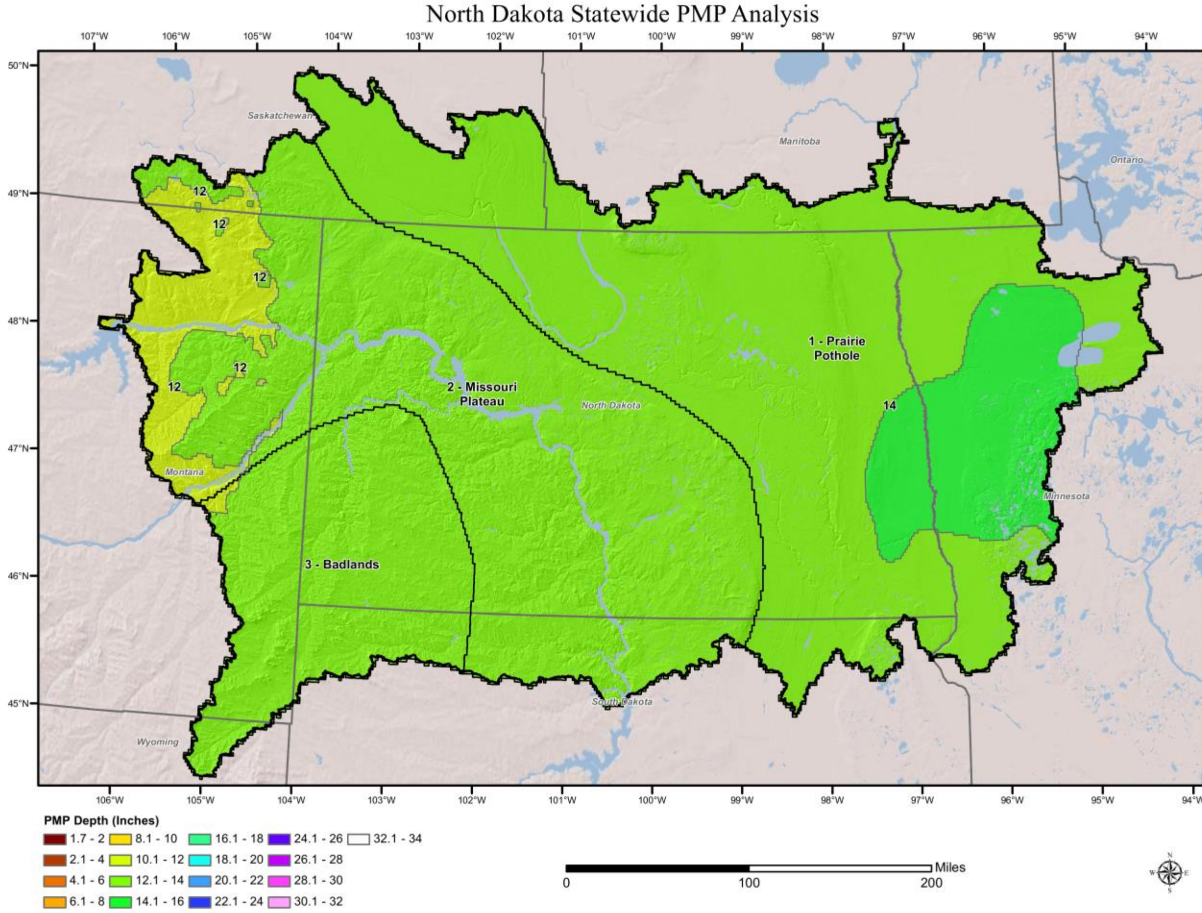
24-Hour General Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



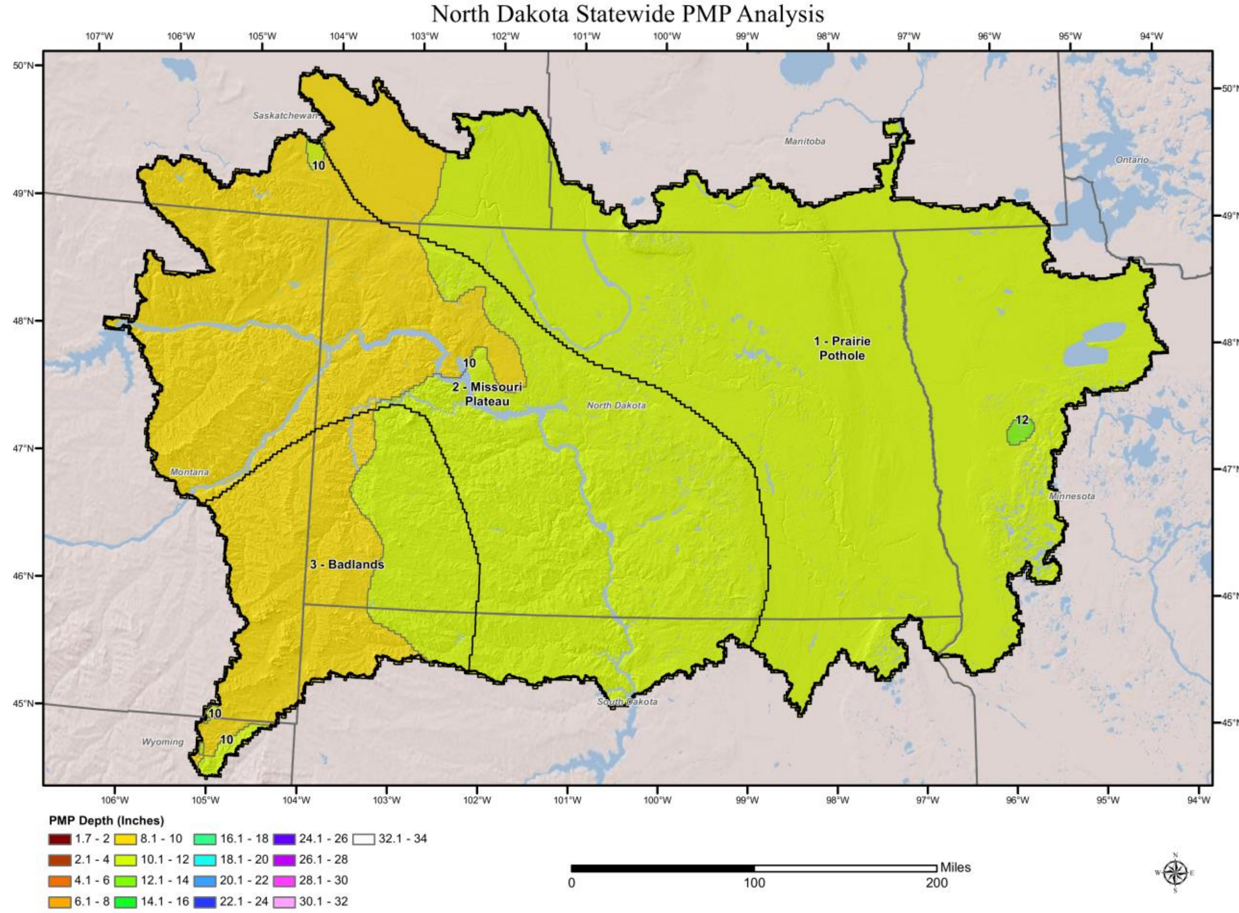
24-Hour General Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



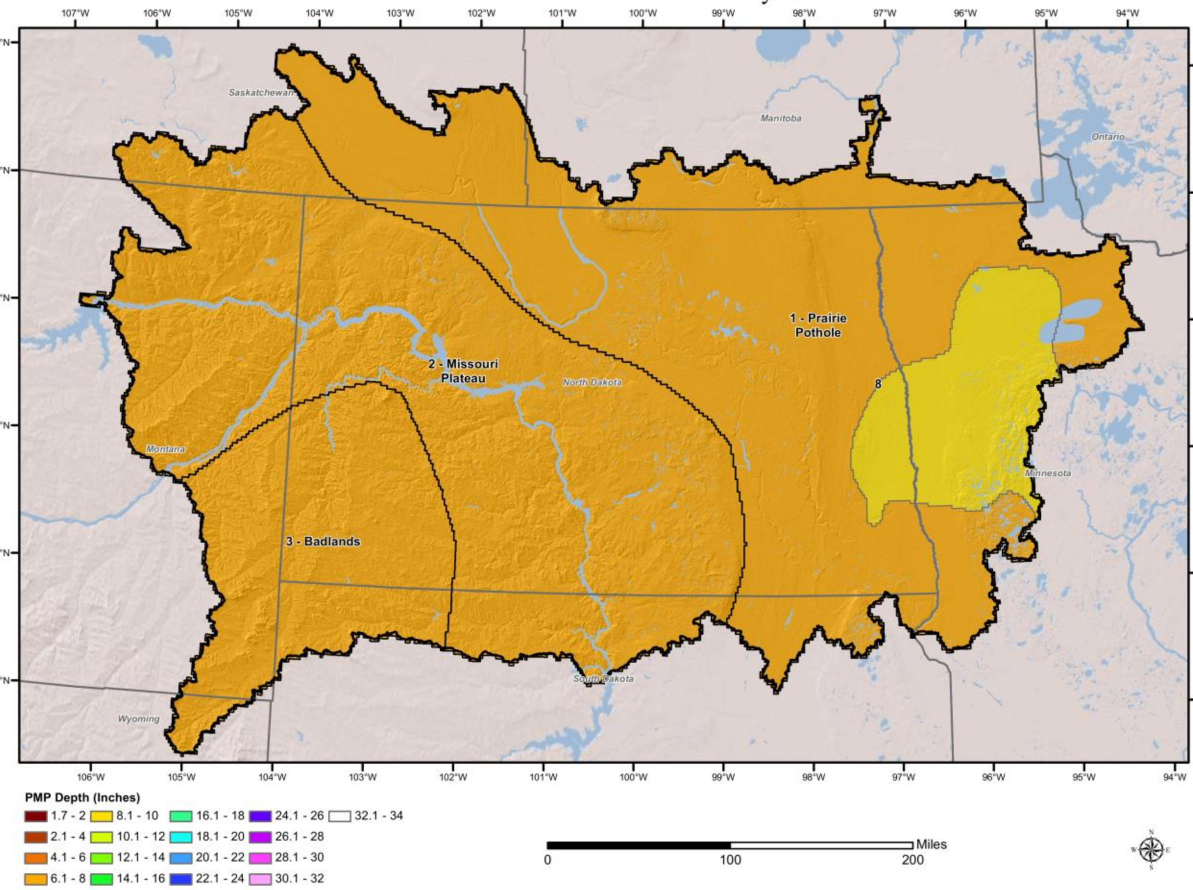
24-Hour General Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis



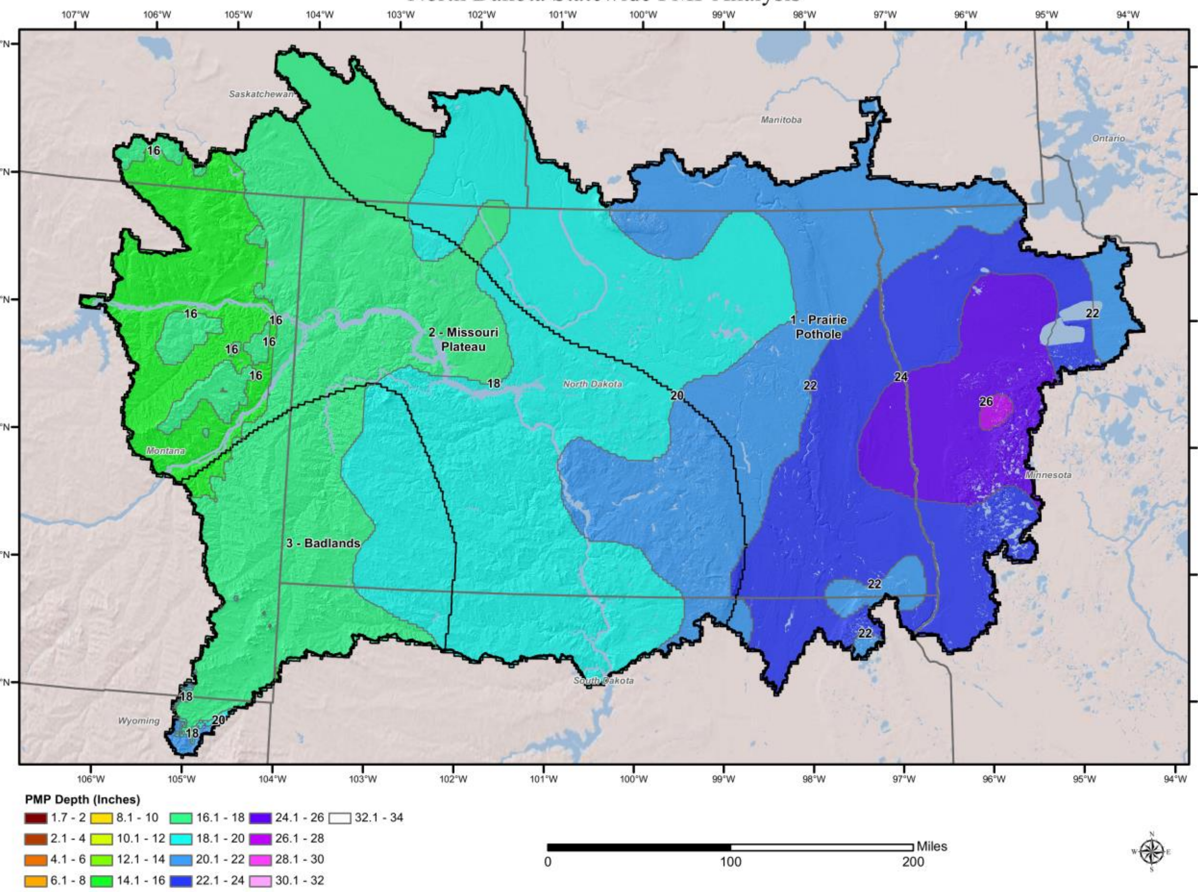
24-Hour General Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis



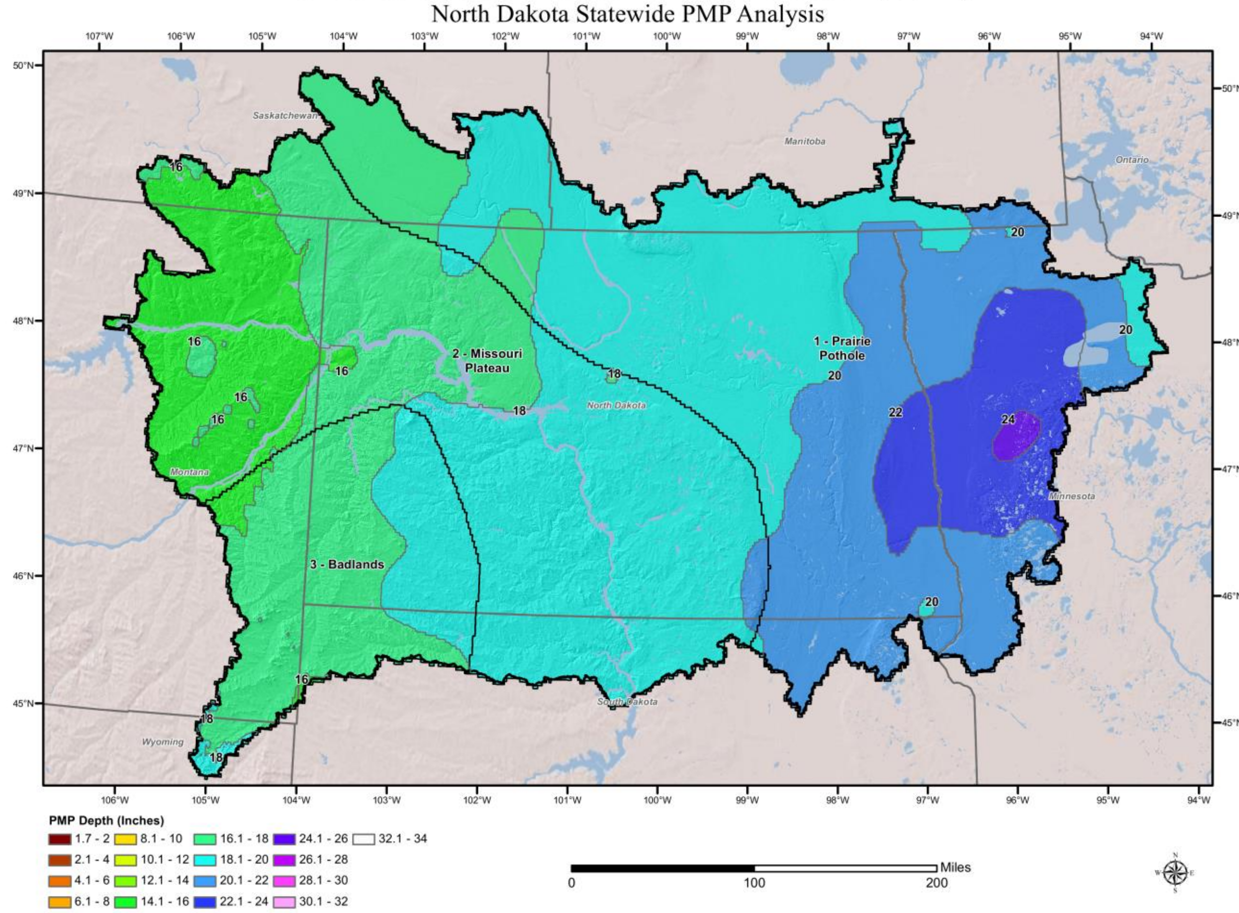
24-Hour General Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis



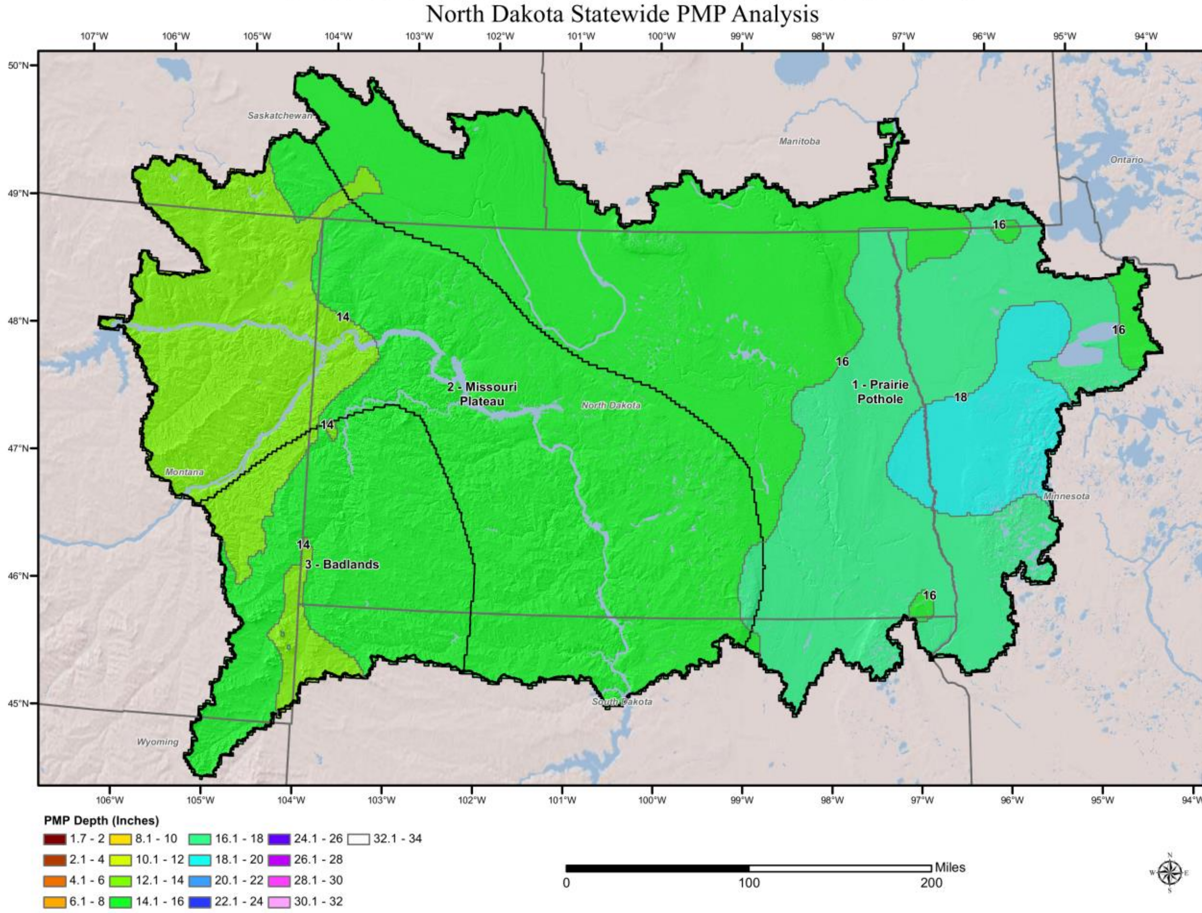
48-Hour General Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



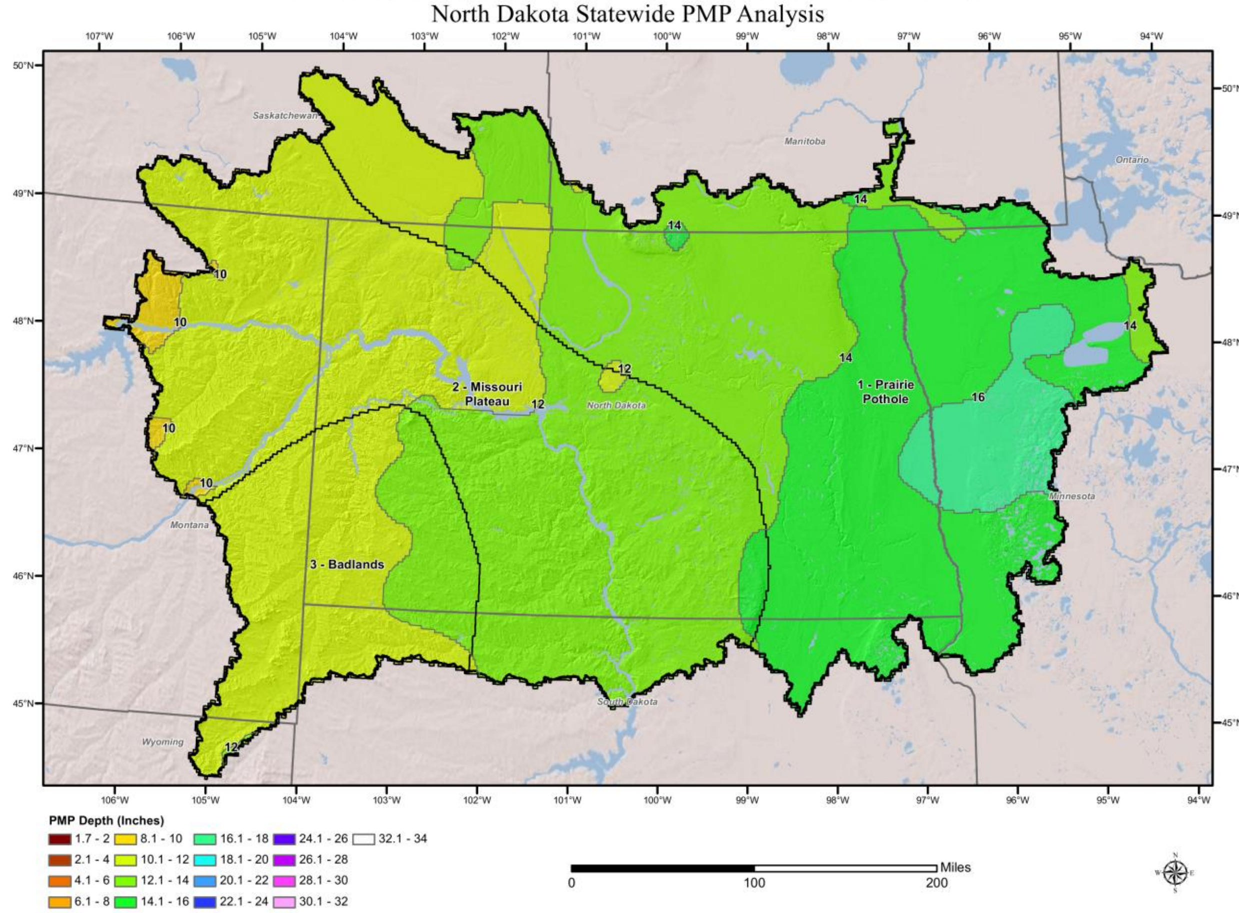
48-Hour General Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



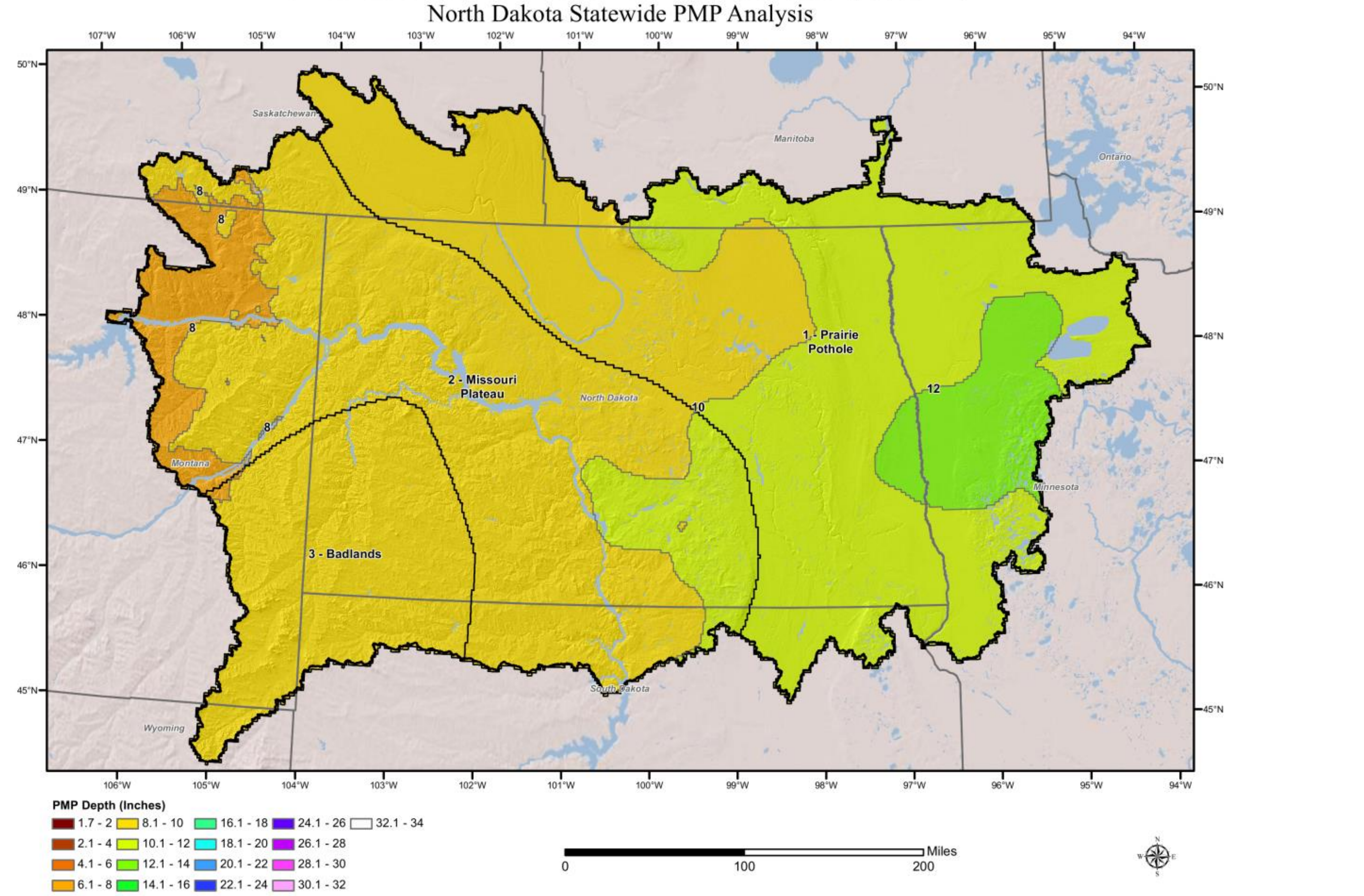
48-Hour General Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis



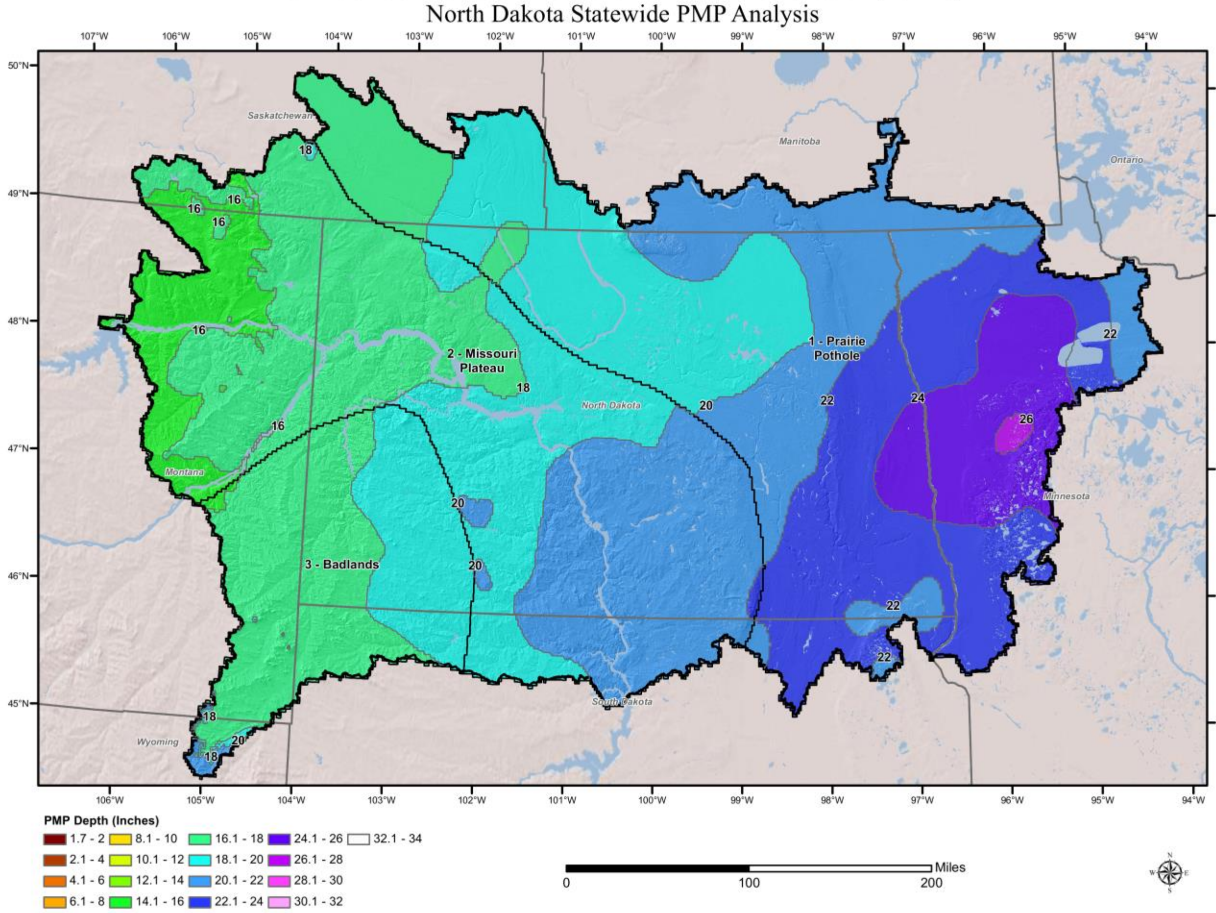
48-Hour General Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis



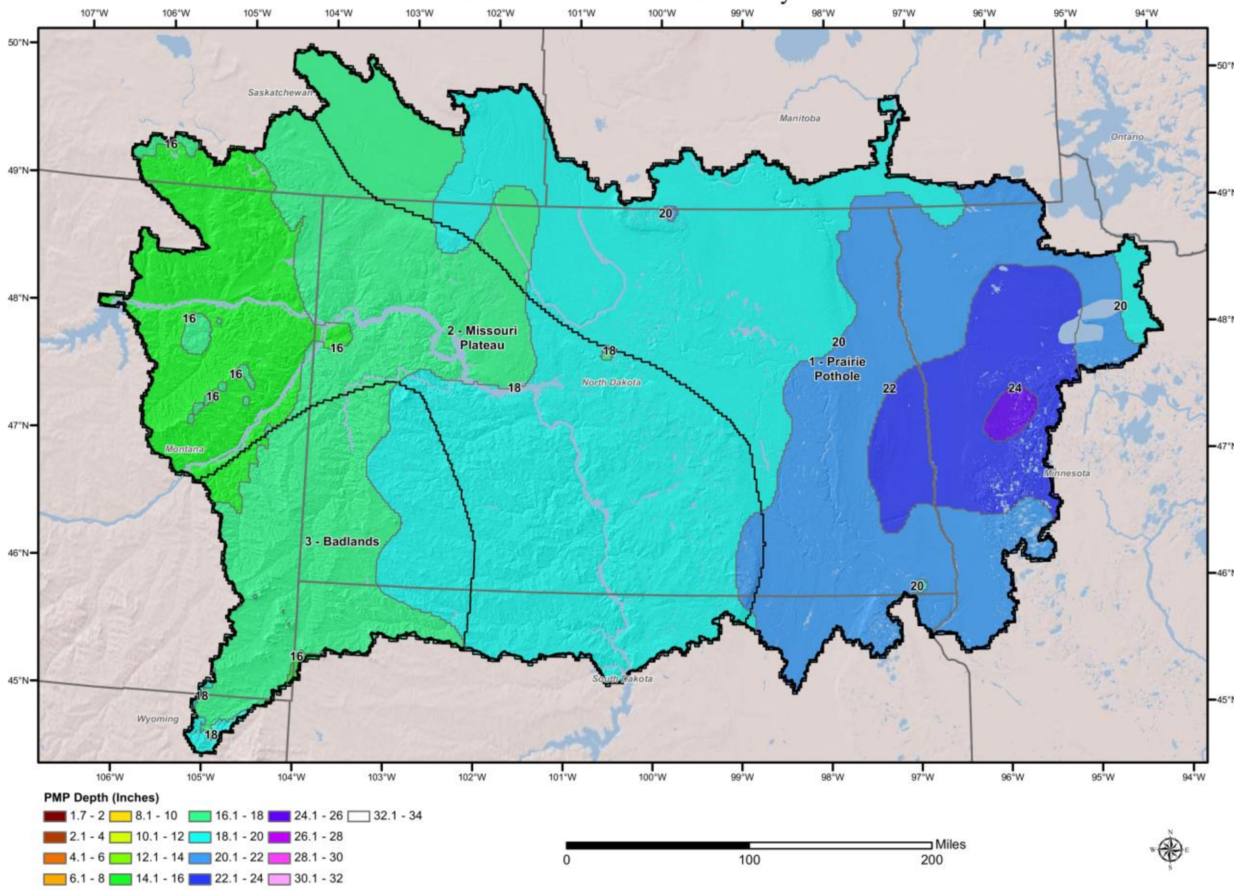
48-Hour General Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis



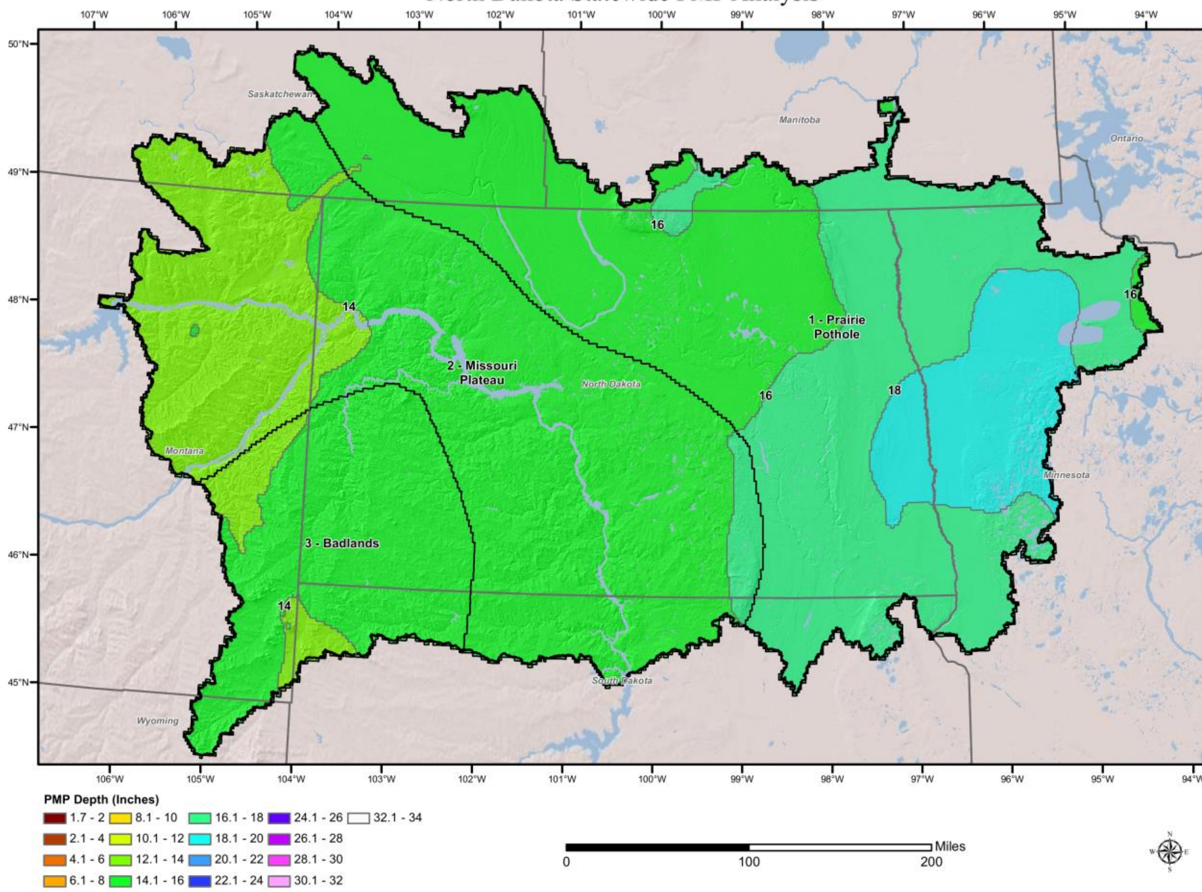
72-Hour General Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



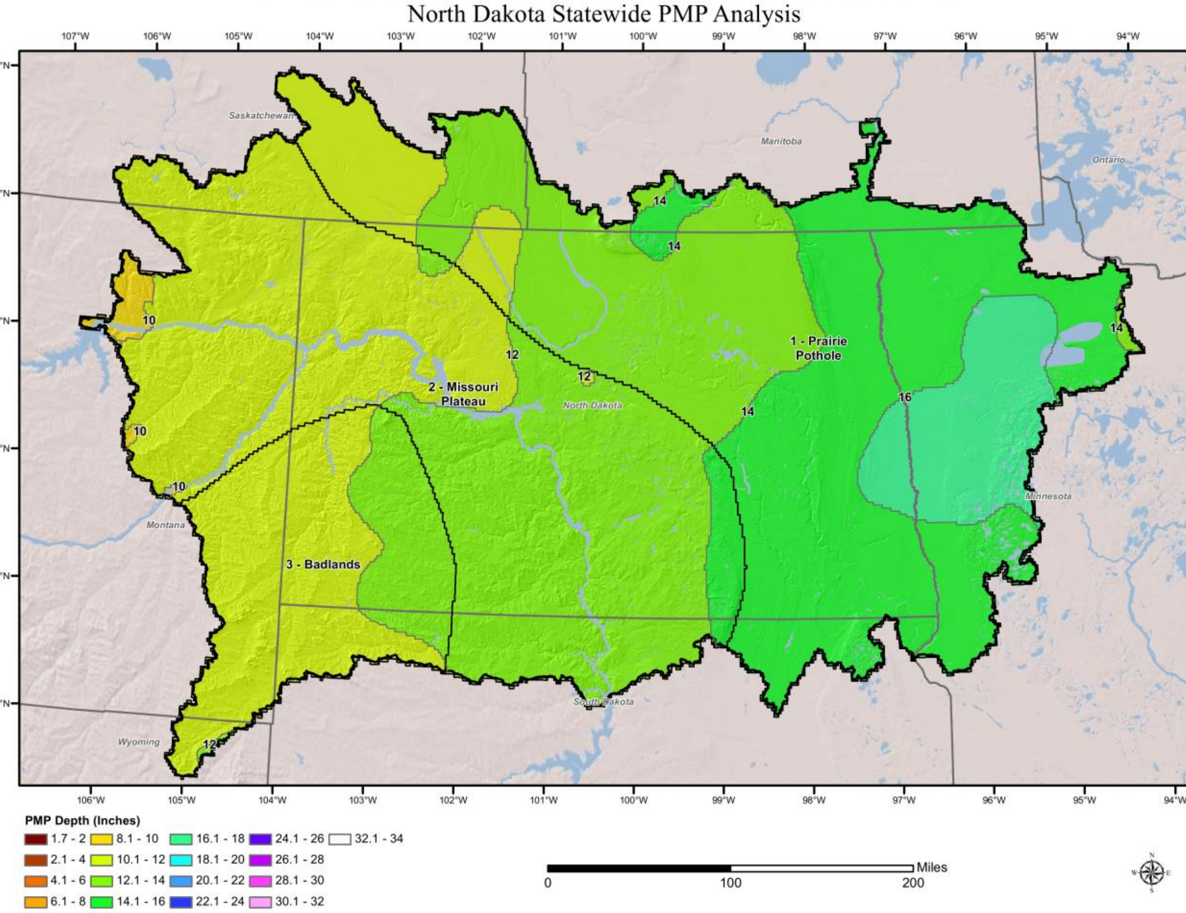
72-Hour General Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



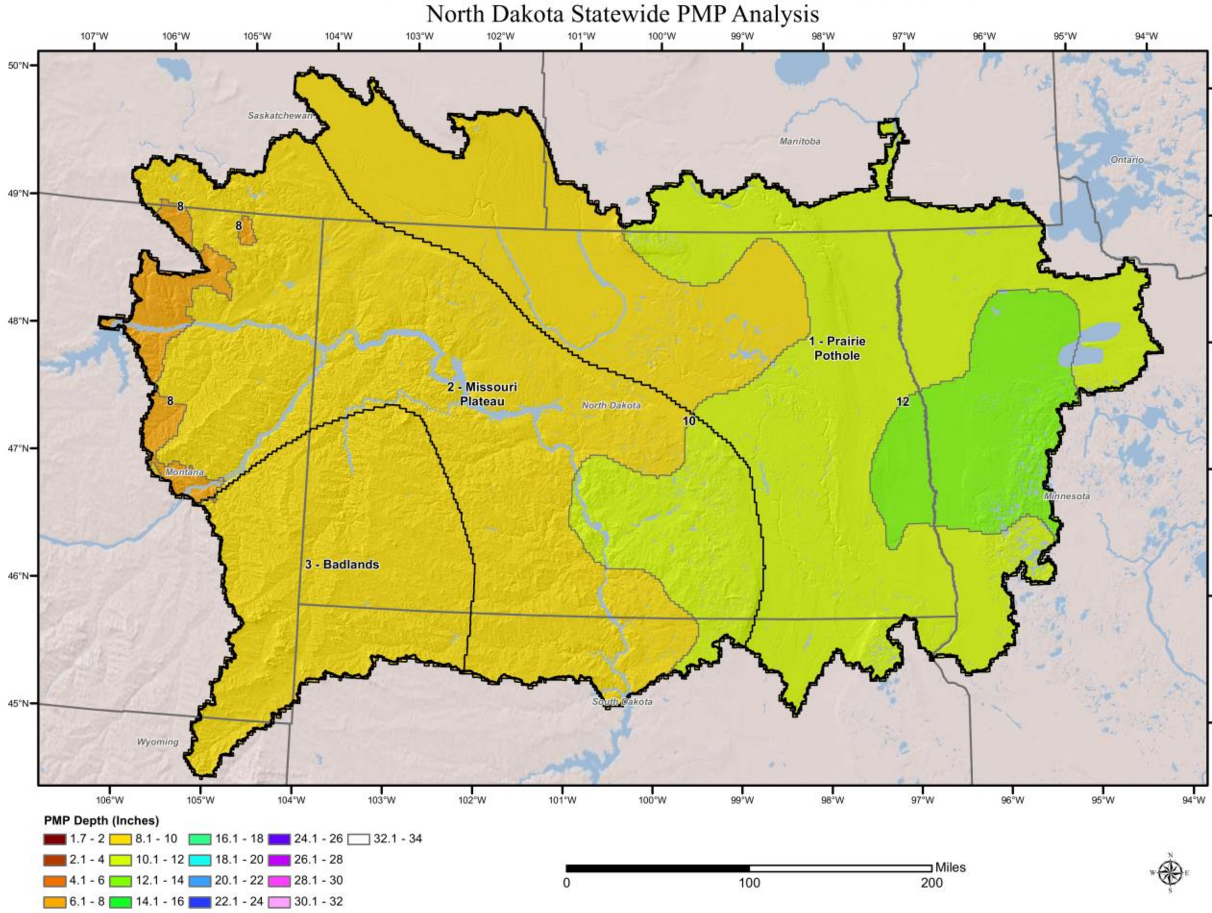
72-Hour General Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis



72-Hour General Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis

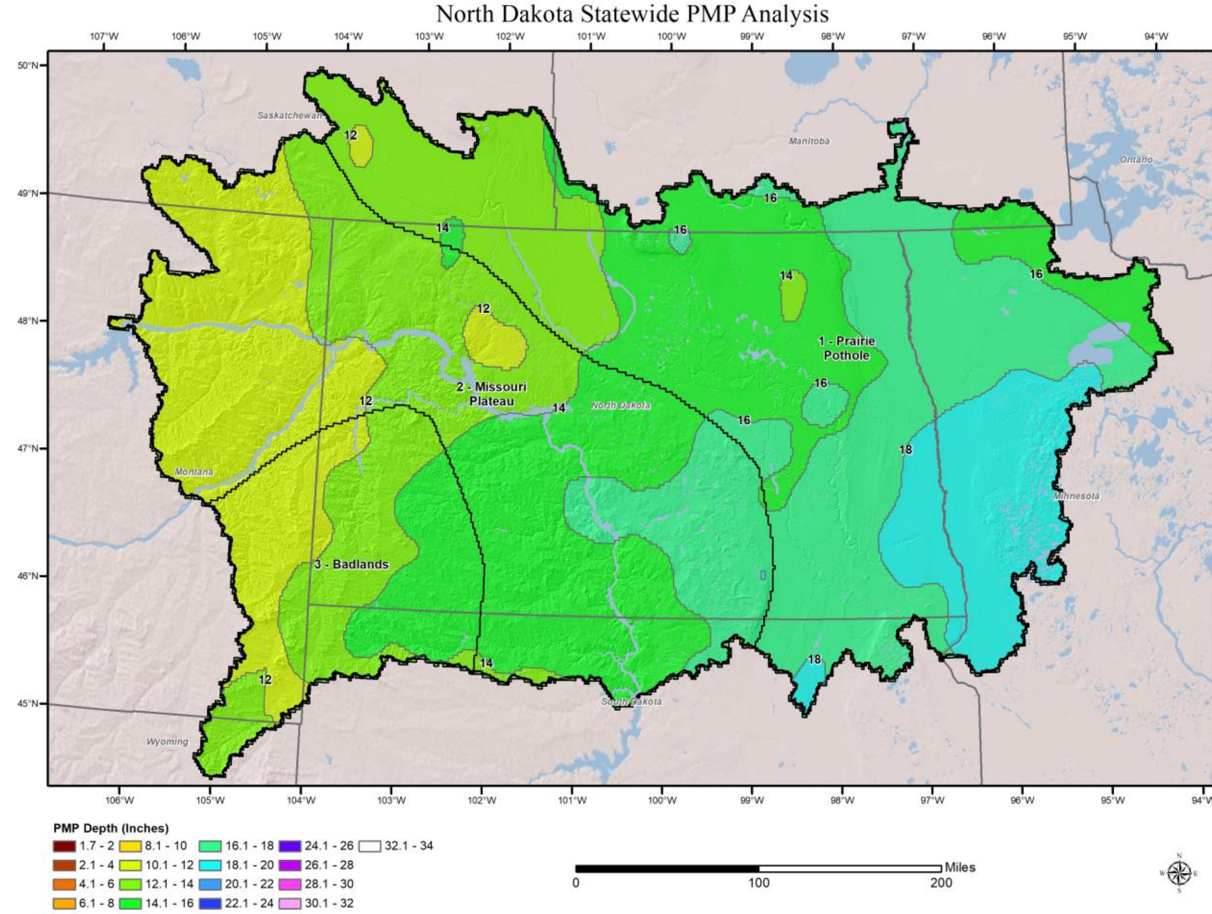


72-Hour General Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis

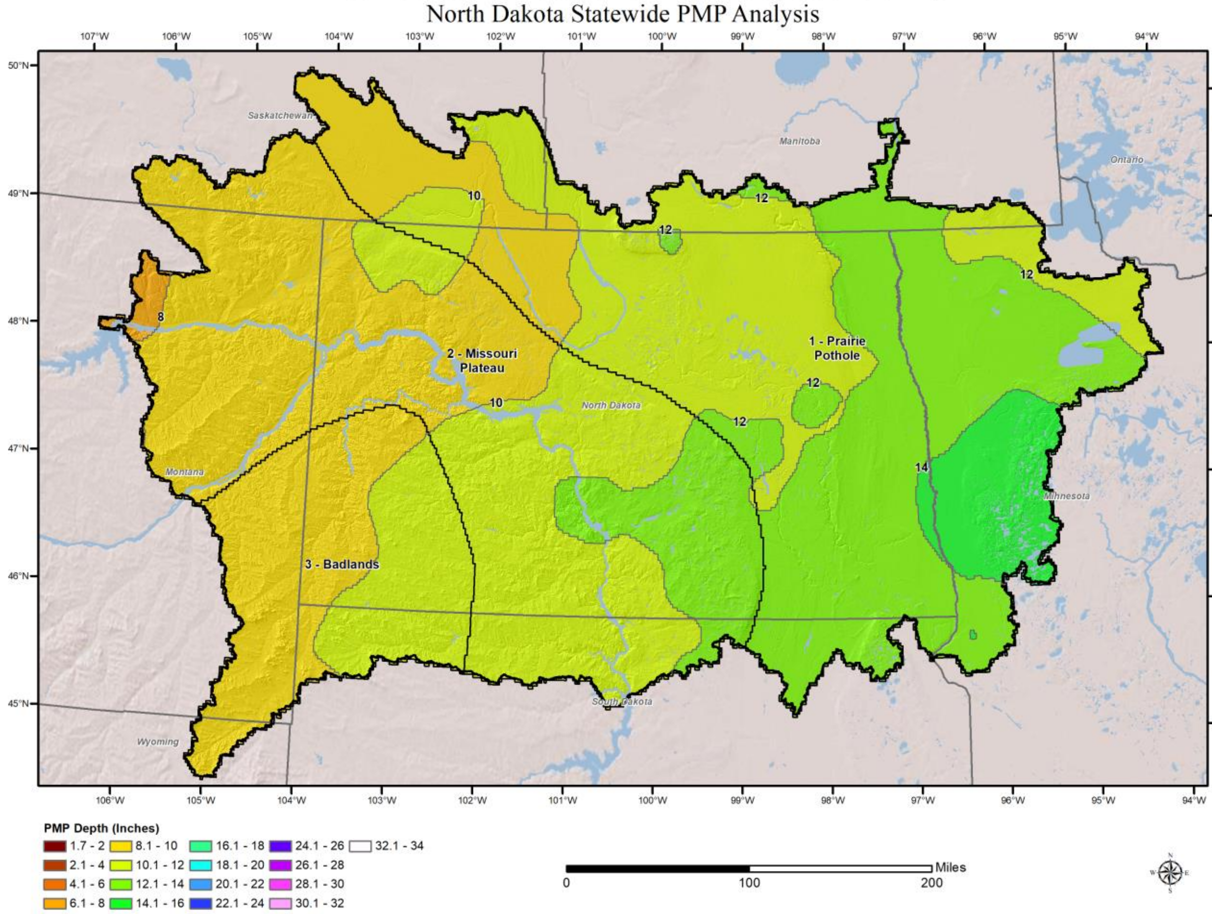


Local Storms

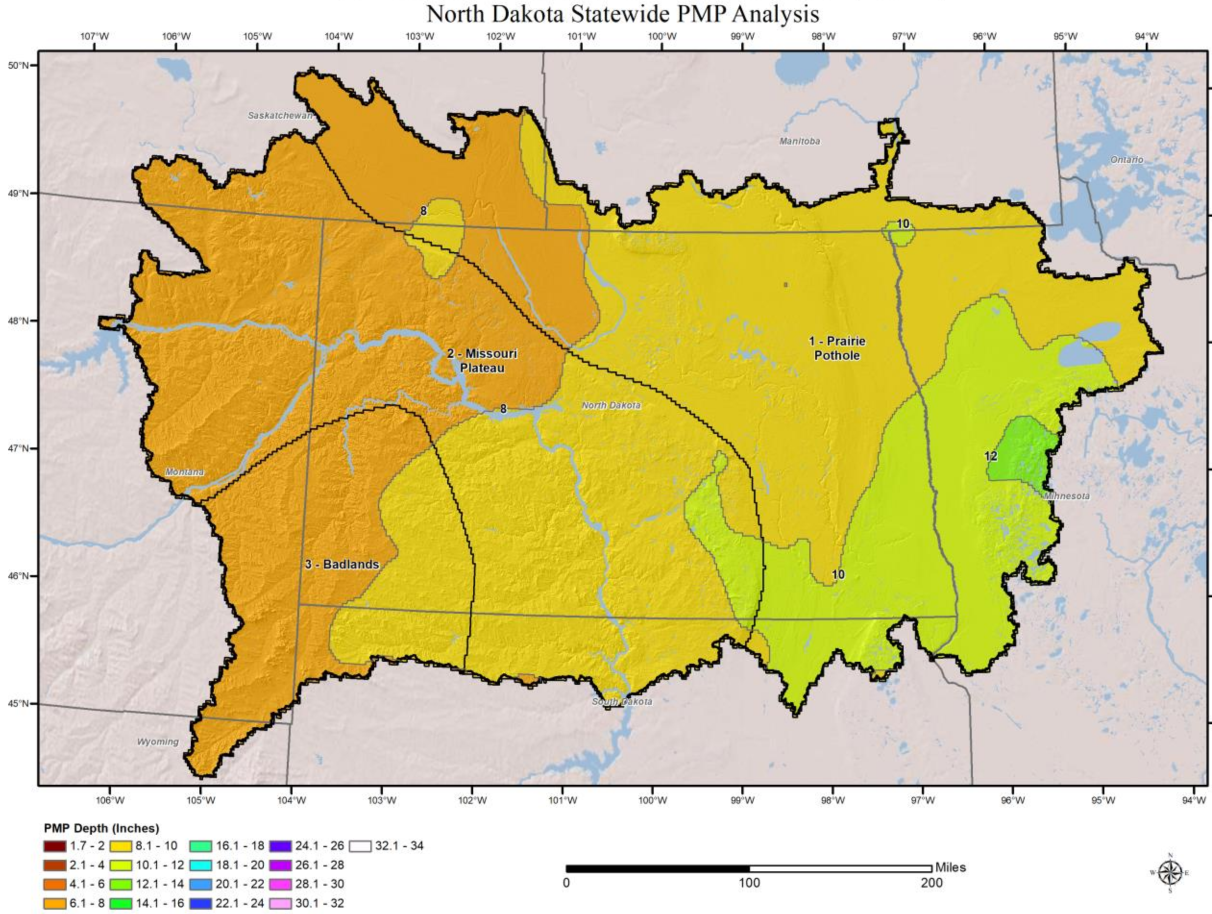
1-Hour Local Storm Probable Maximum Precipitation (1 mi²)
North Dakota Statewide PMP Analysis



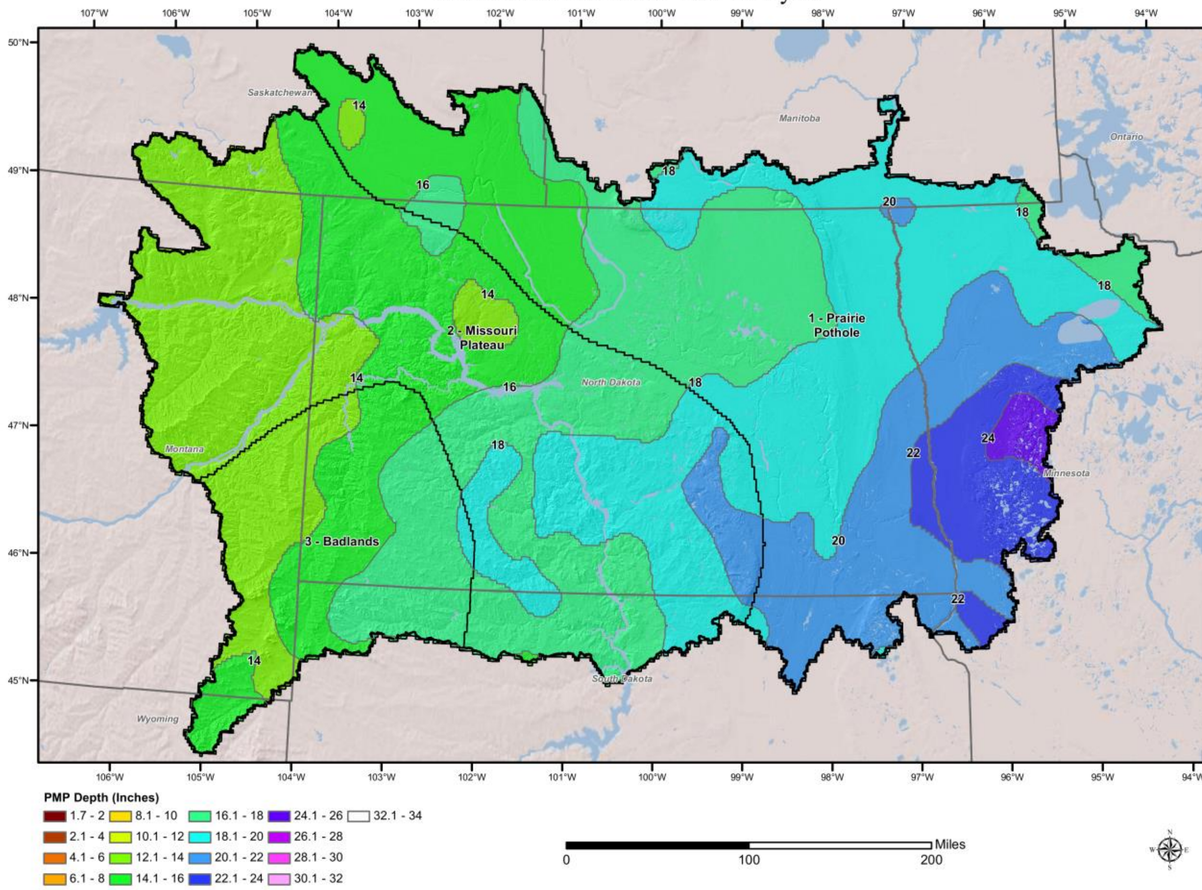
1-Hour Local Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



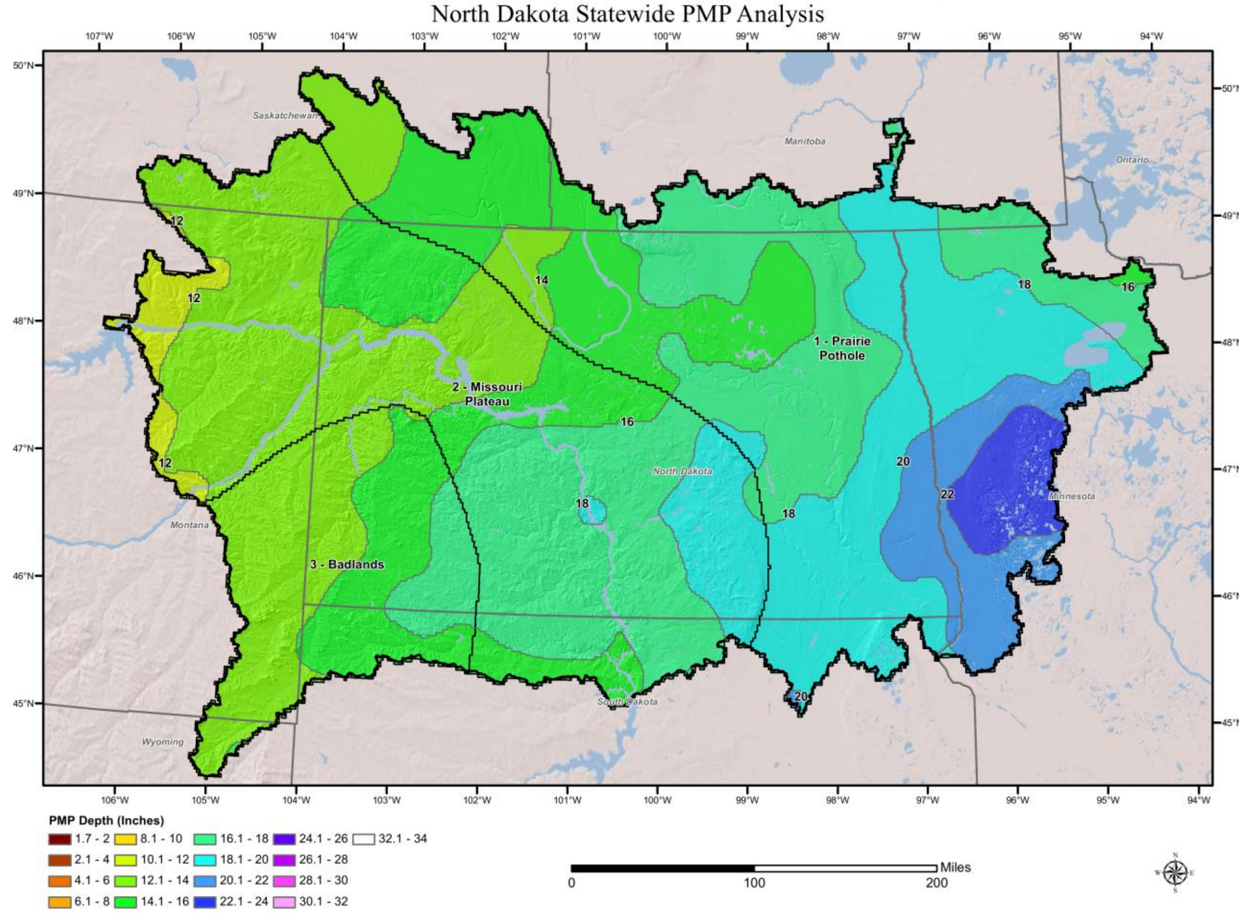
1-Hour Local Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



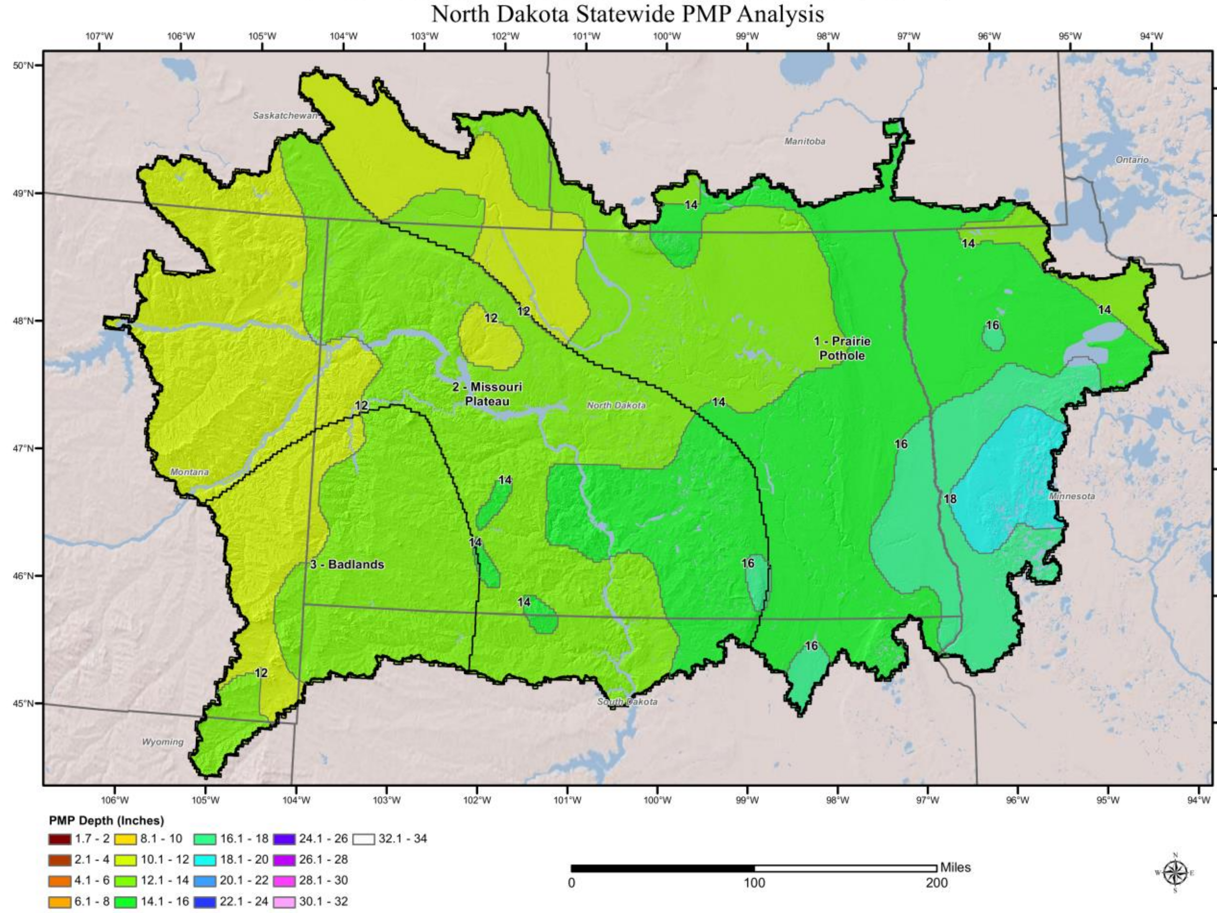
3-Hour Local Storm Probable Maximum Precipitation (1 mi²)
North Dakota Statewide PMP Analysis



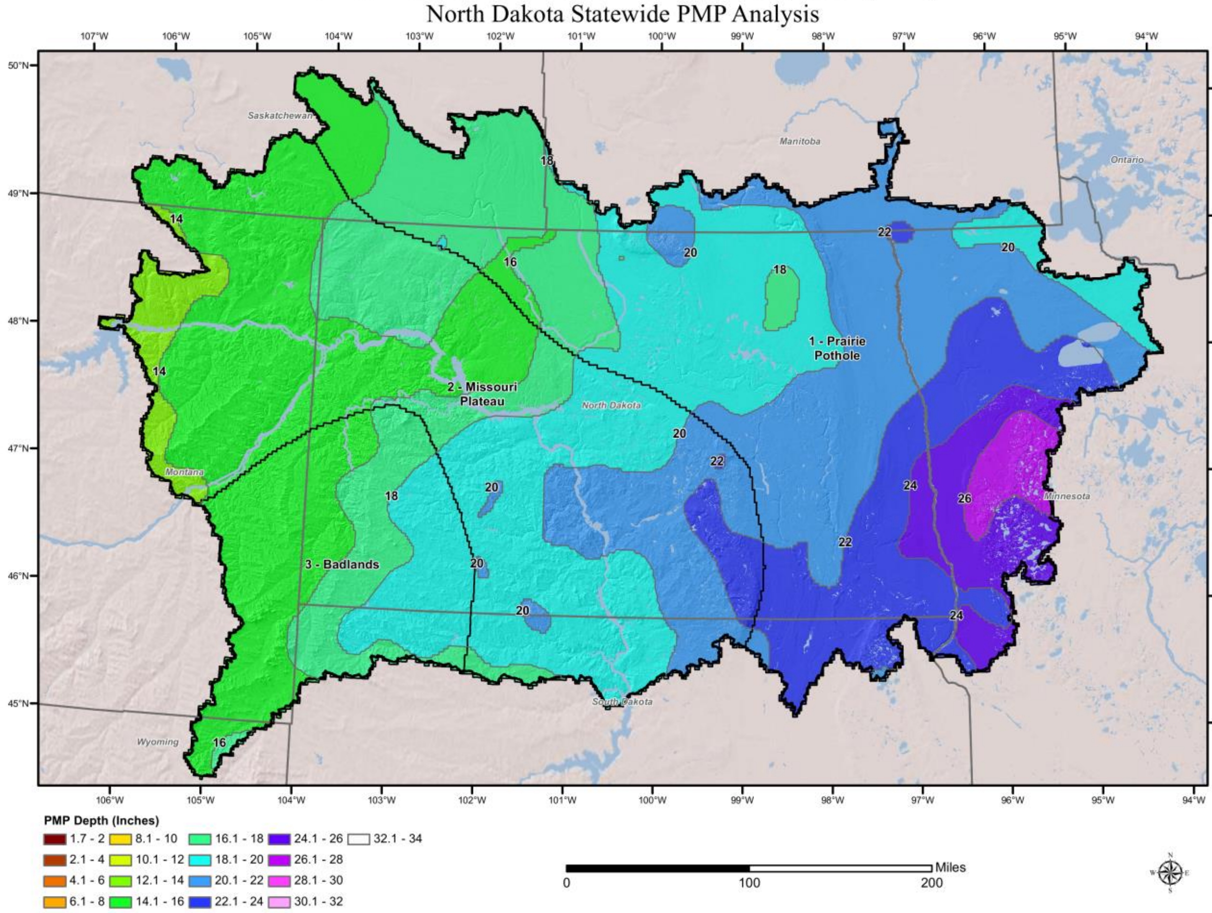
3-Hour Local Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



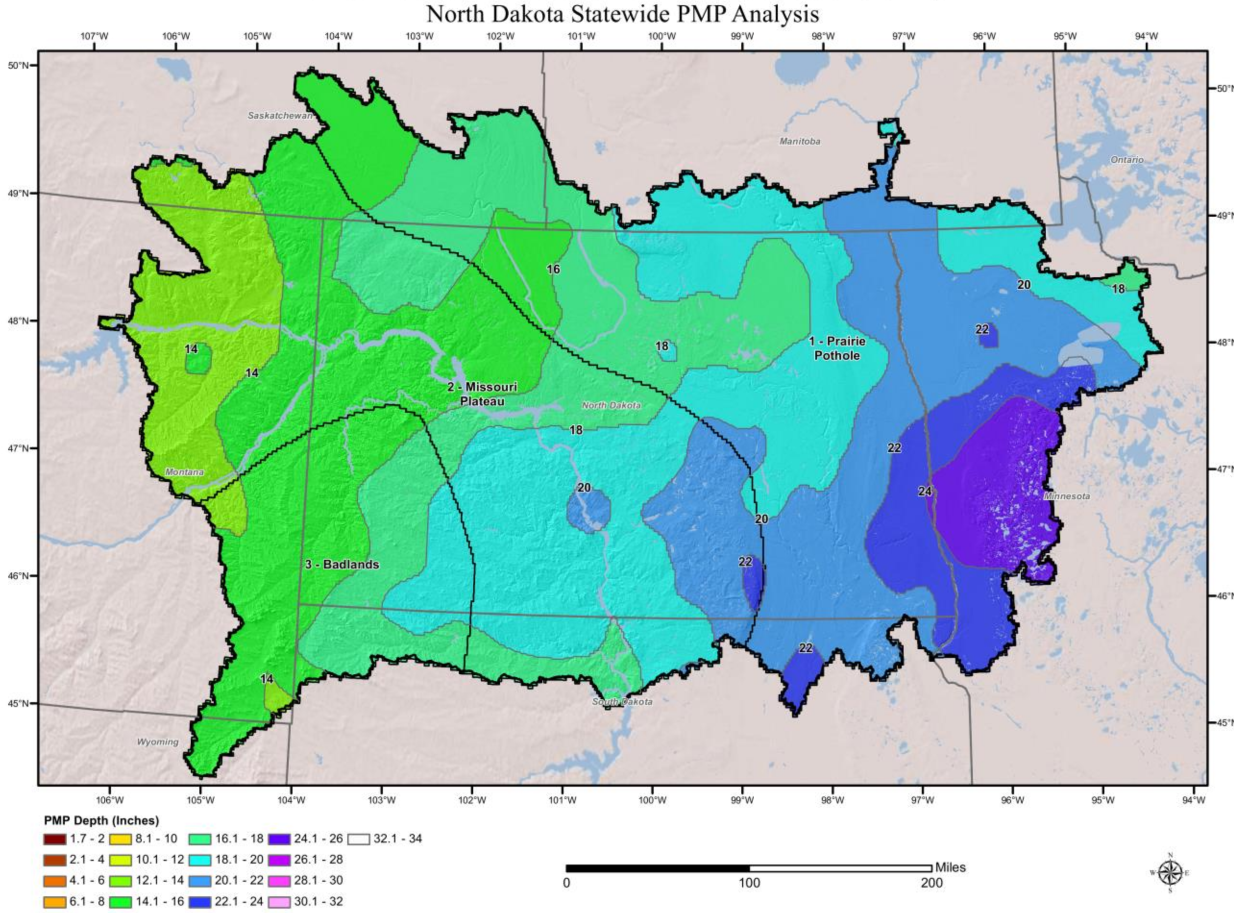
3-Hour Local Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



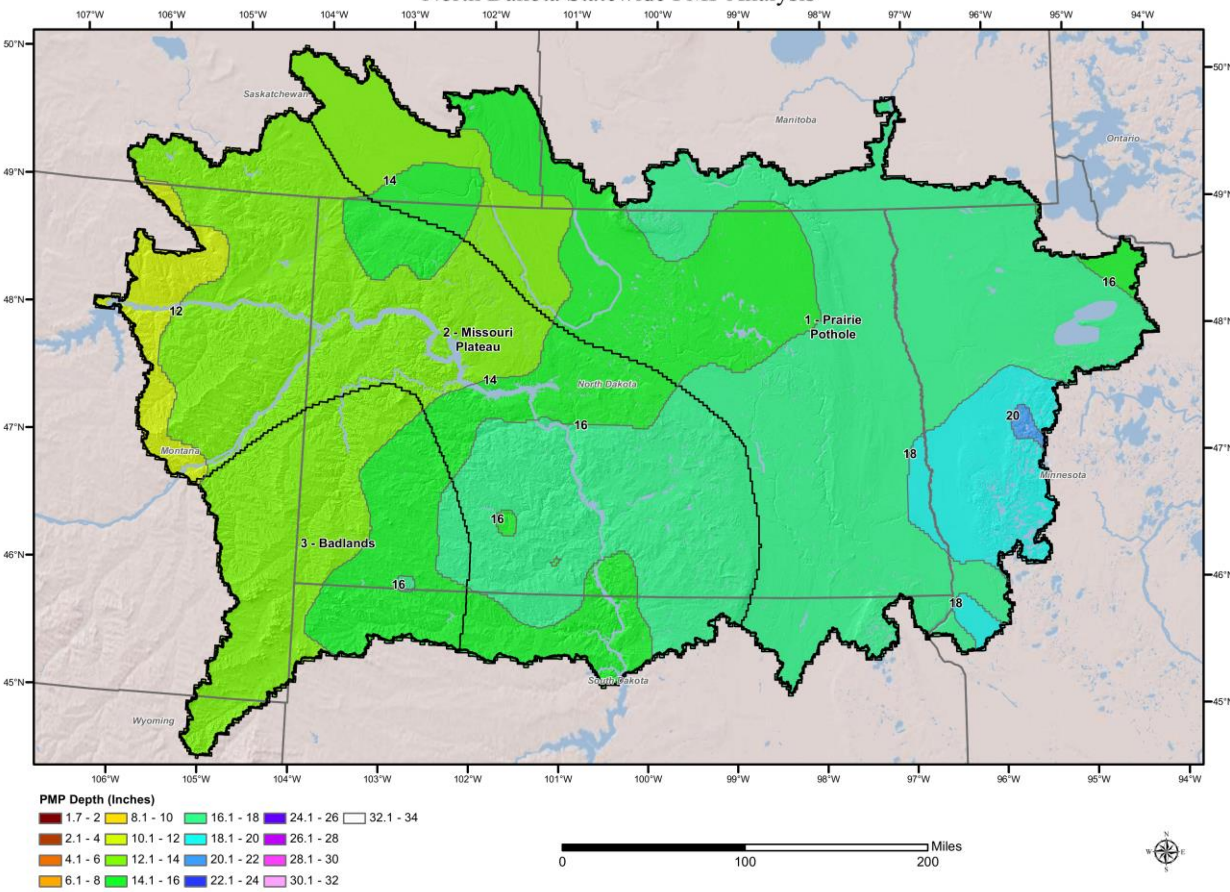
6-Hour Local Storm Probable Maximum Precipitation (1 mi²)
North Dakota Statewide PMP Analysis



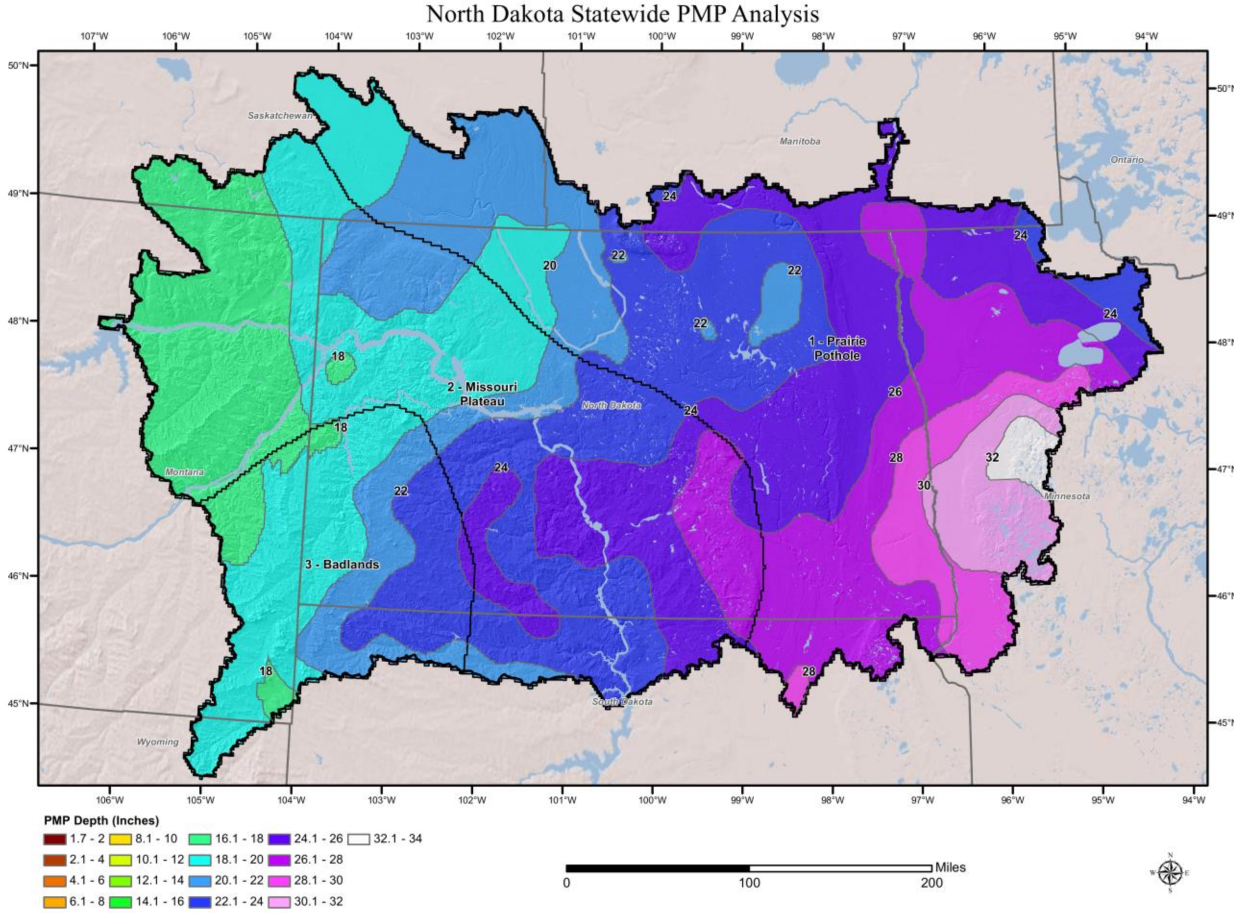
6-Hour Local Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



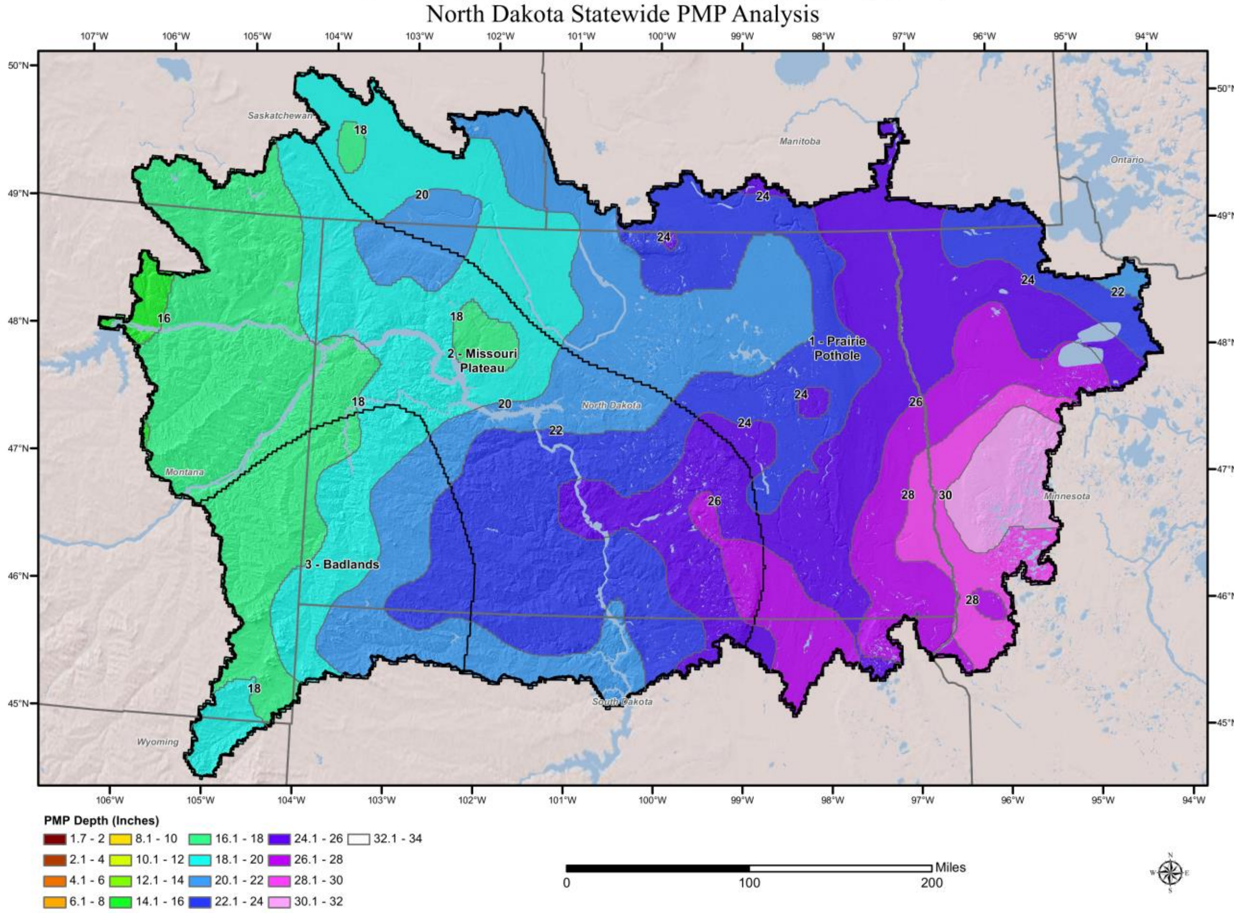
6-Hour Local Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



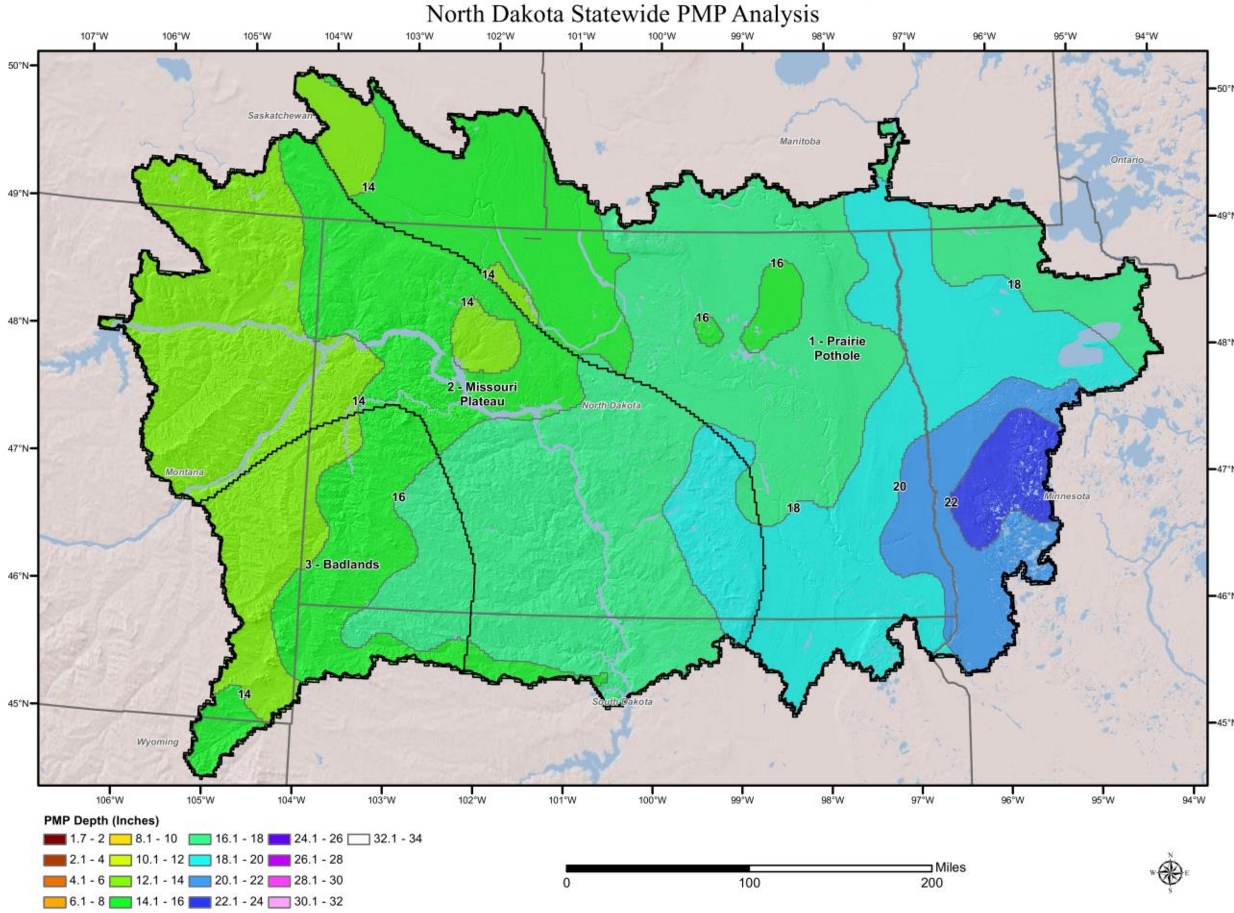
12-Hour Local Storm Probable Maximum Precipitation (1 mi²)
North Dakota Statewide PMP Analysis



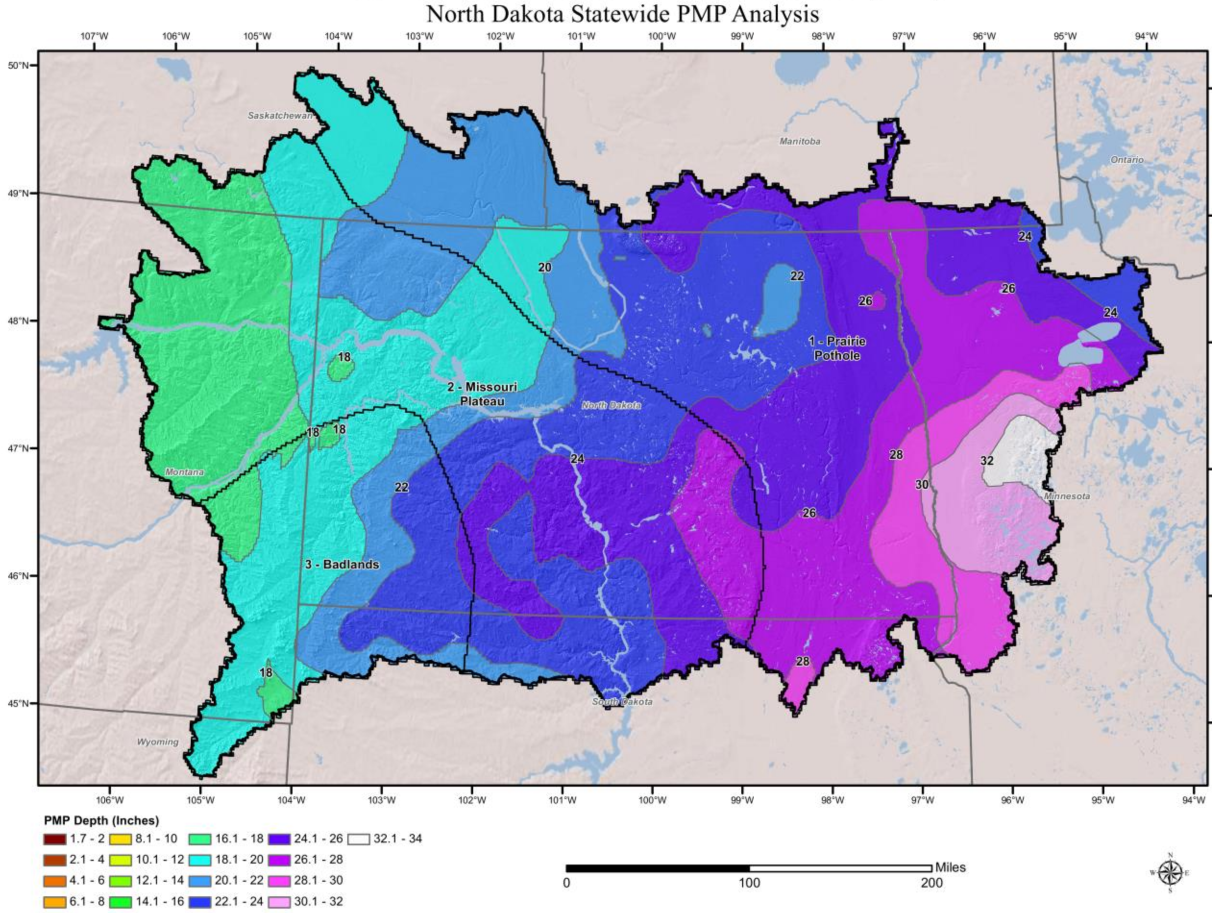
12-Hour Local Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



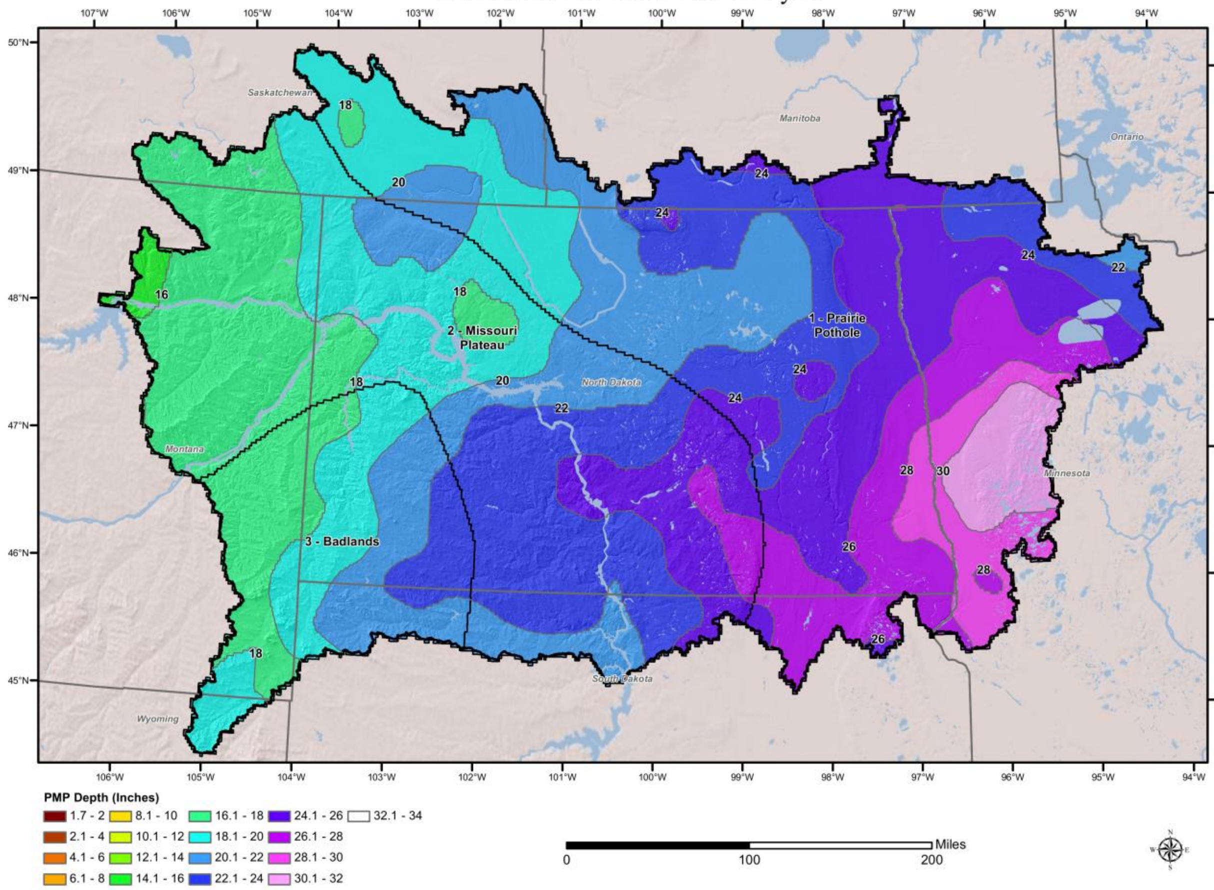
12-Hour Local Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



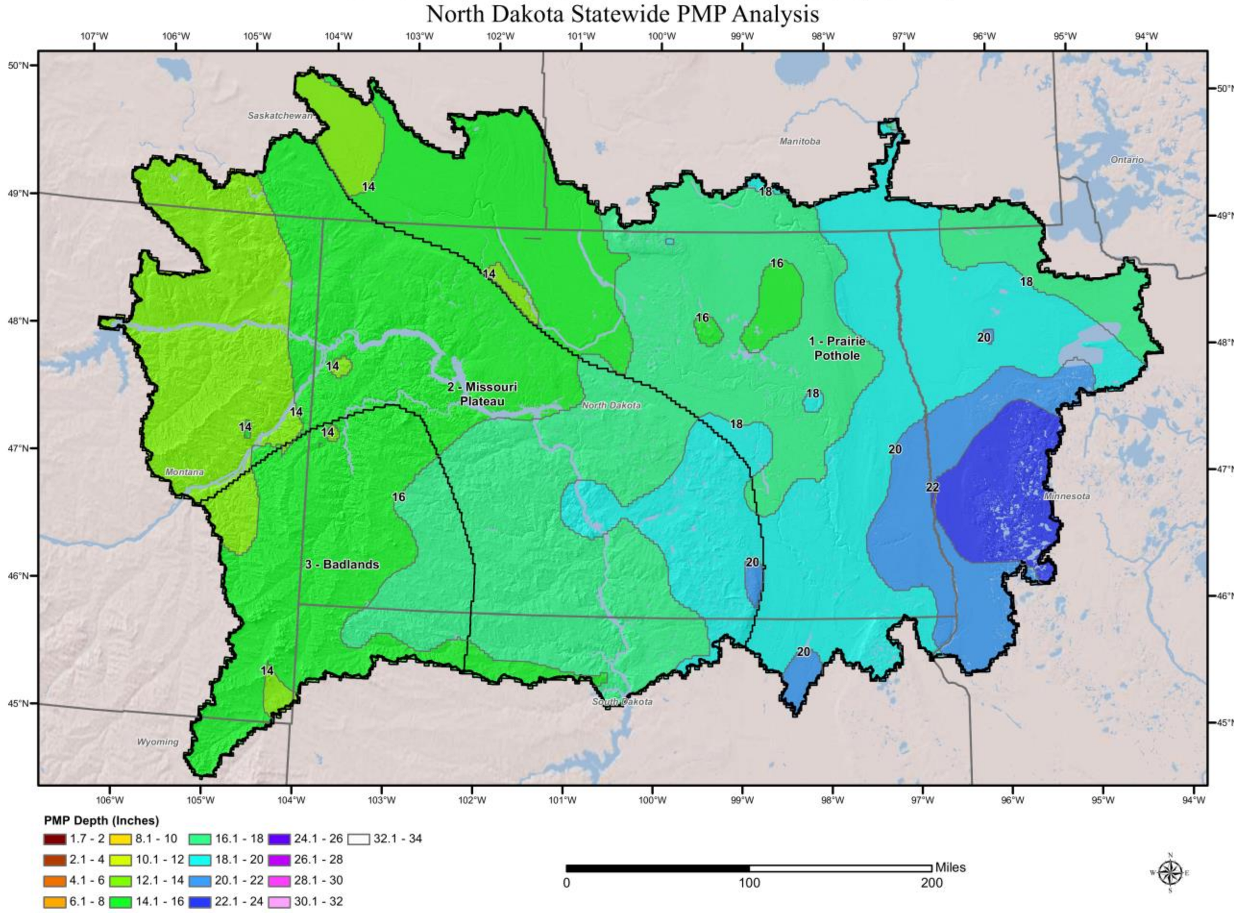
24-Hour Local Storm Probable Maximum Precipitation (1 mi²)
North Dakota Statewide PMP Analysis



24-Hour Local Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis

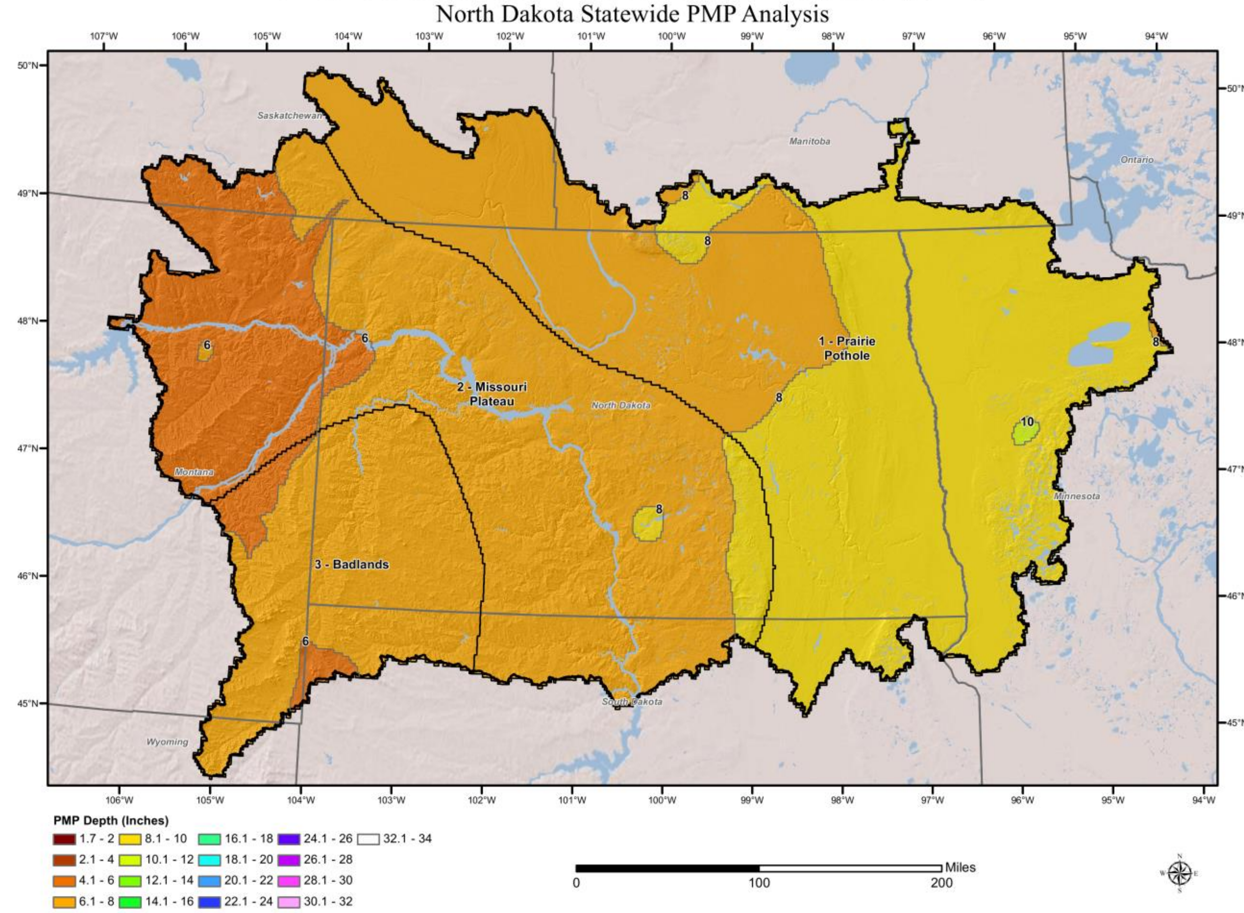


24-Hour Local Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis

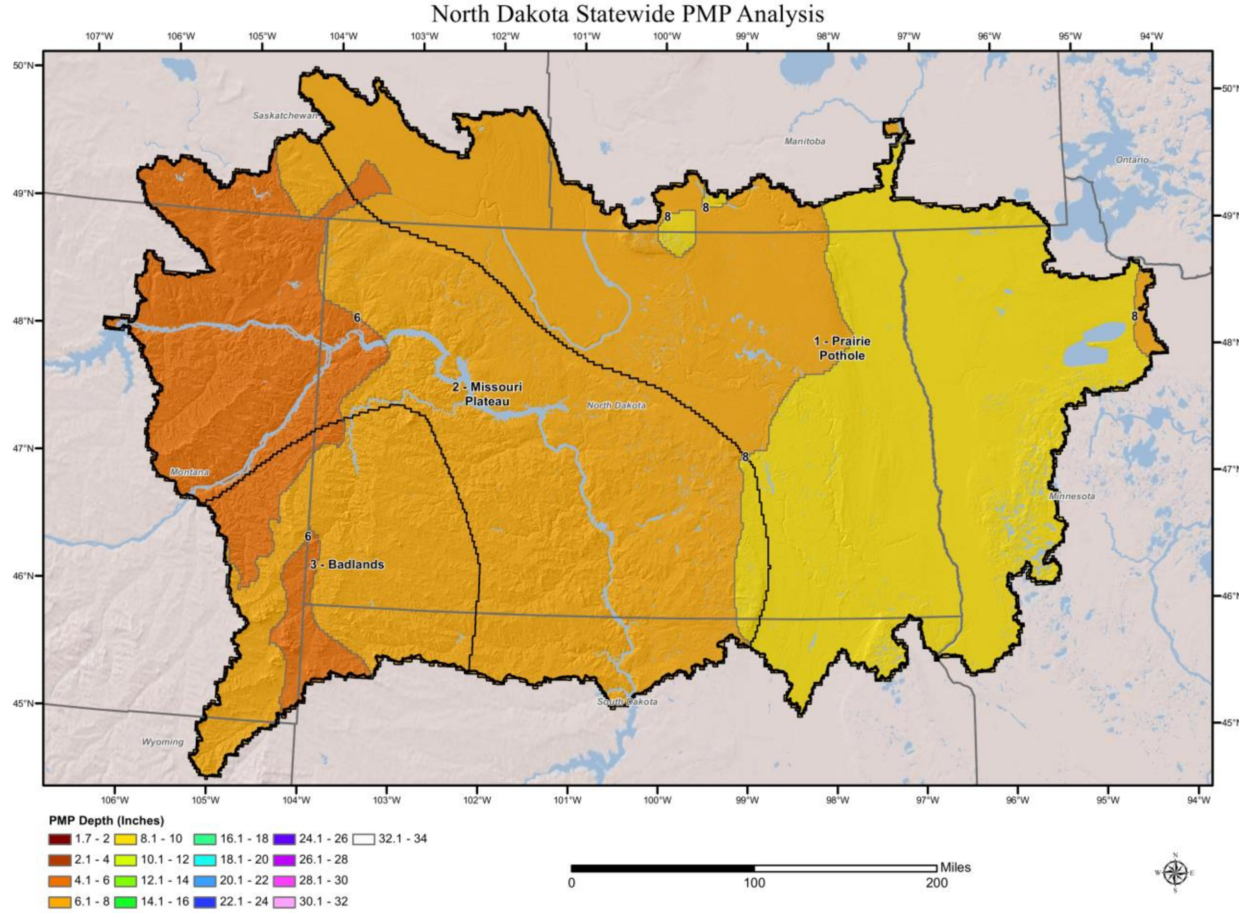


Cool Season Storms

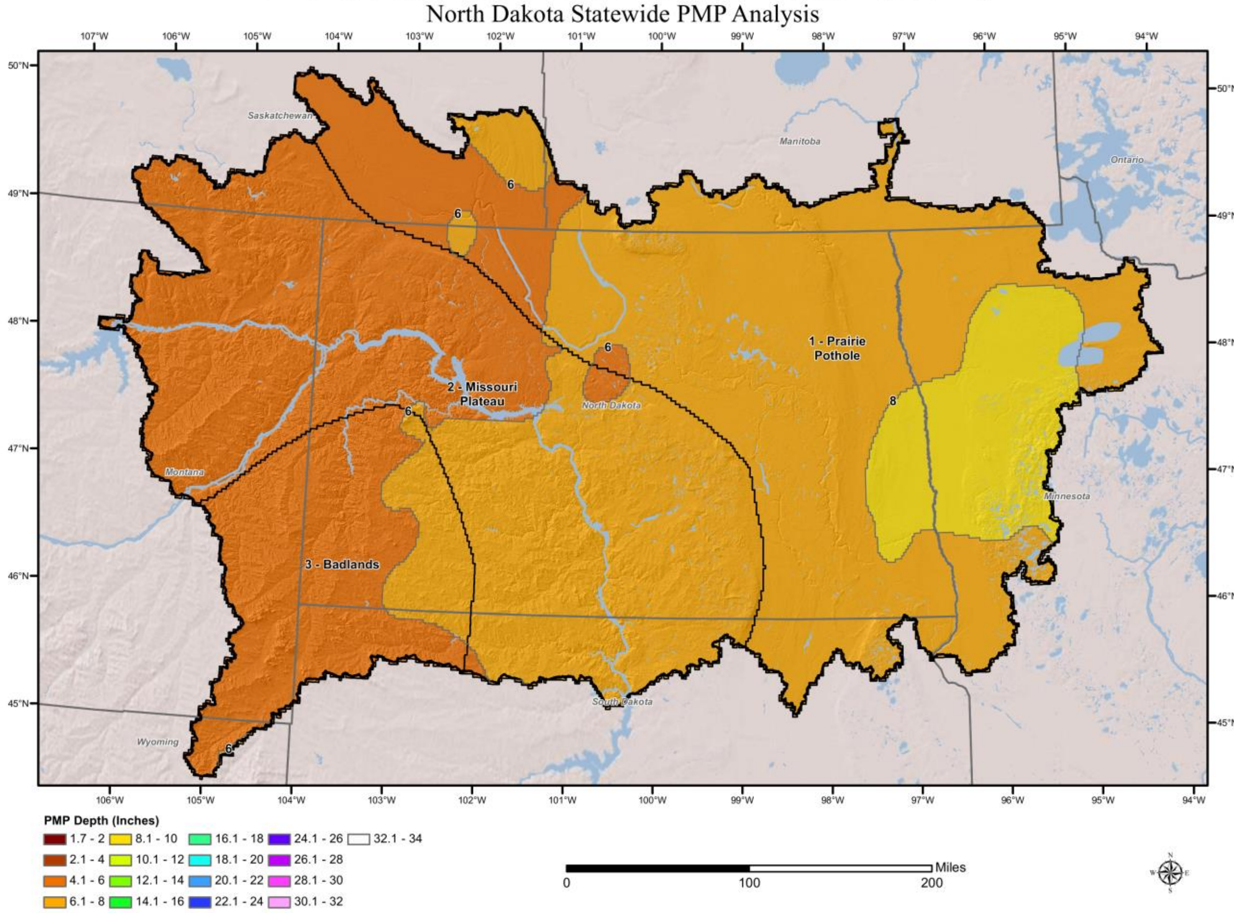
6-Hour Cool Season Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



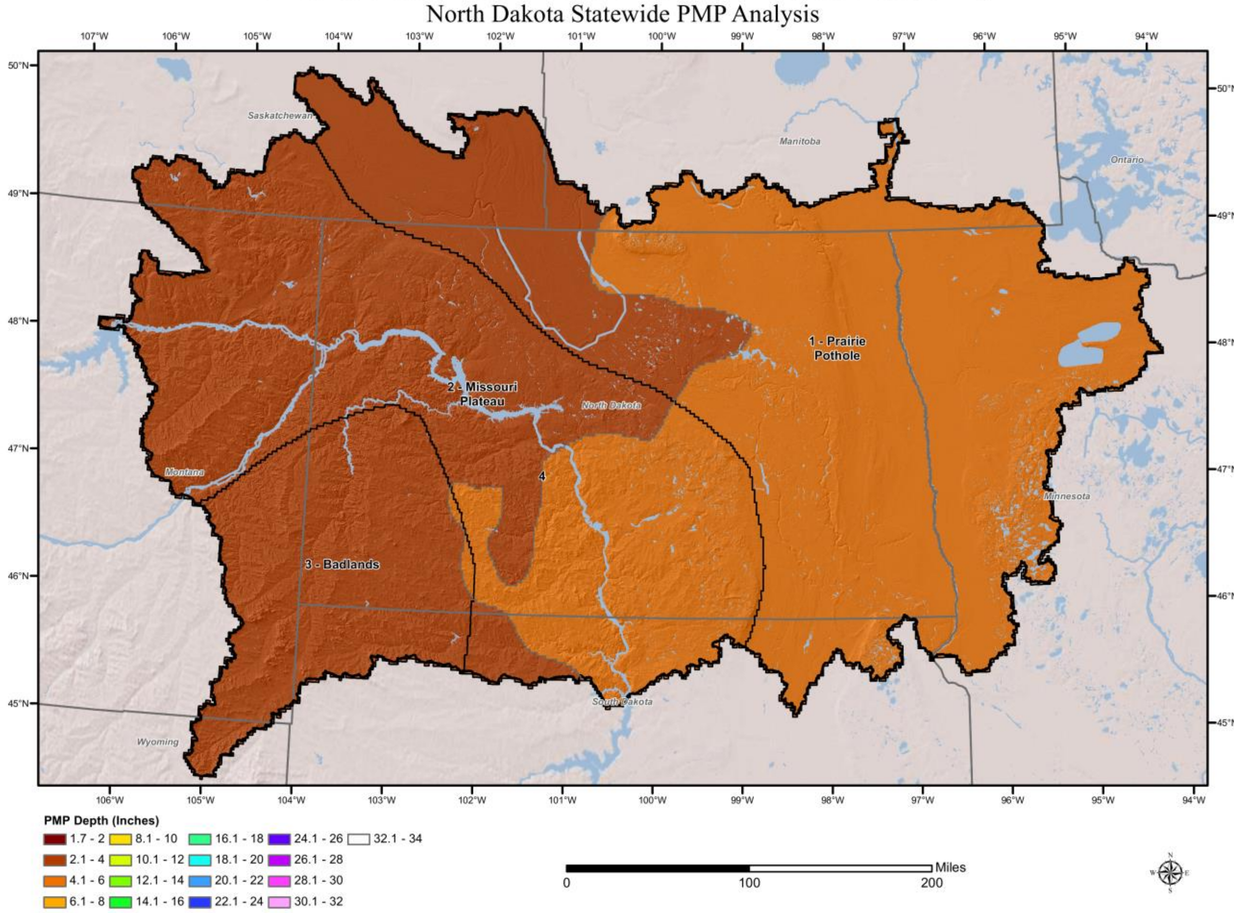
6-Hour Cool Season Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



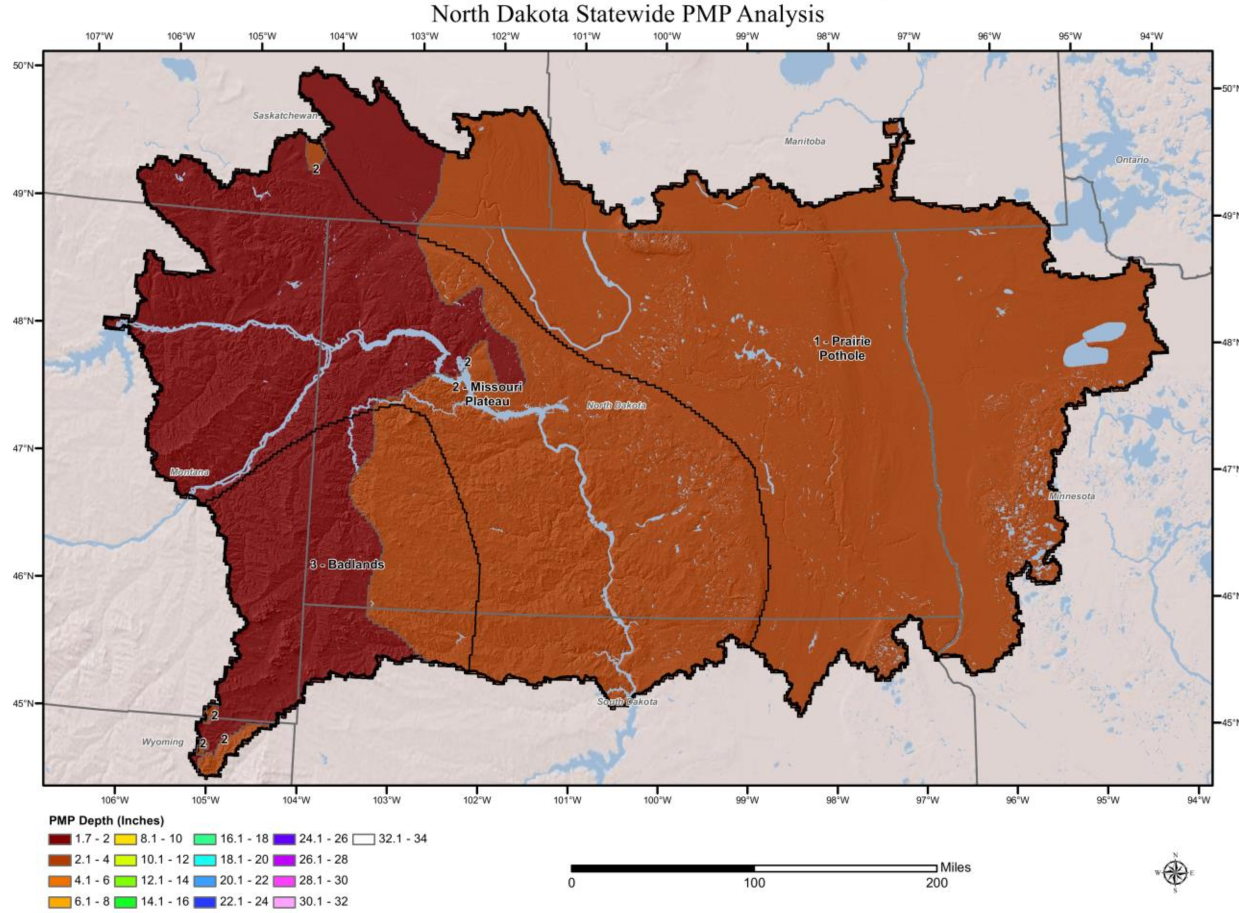
6-Hour Cool Season Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis



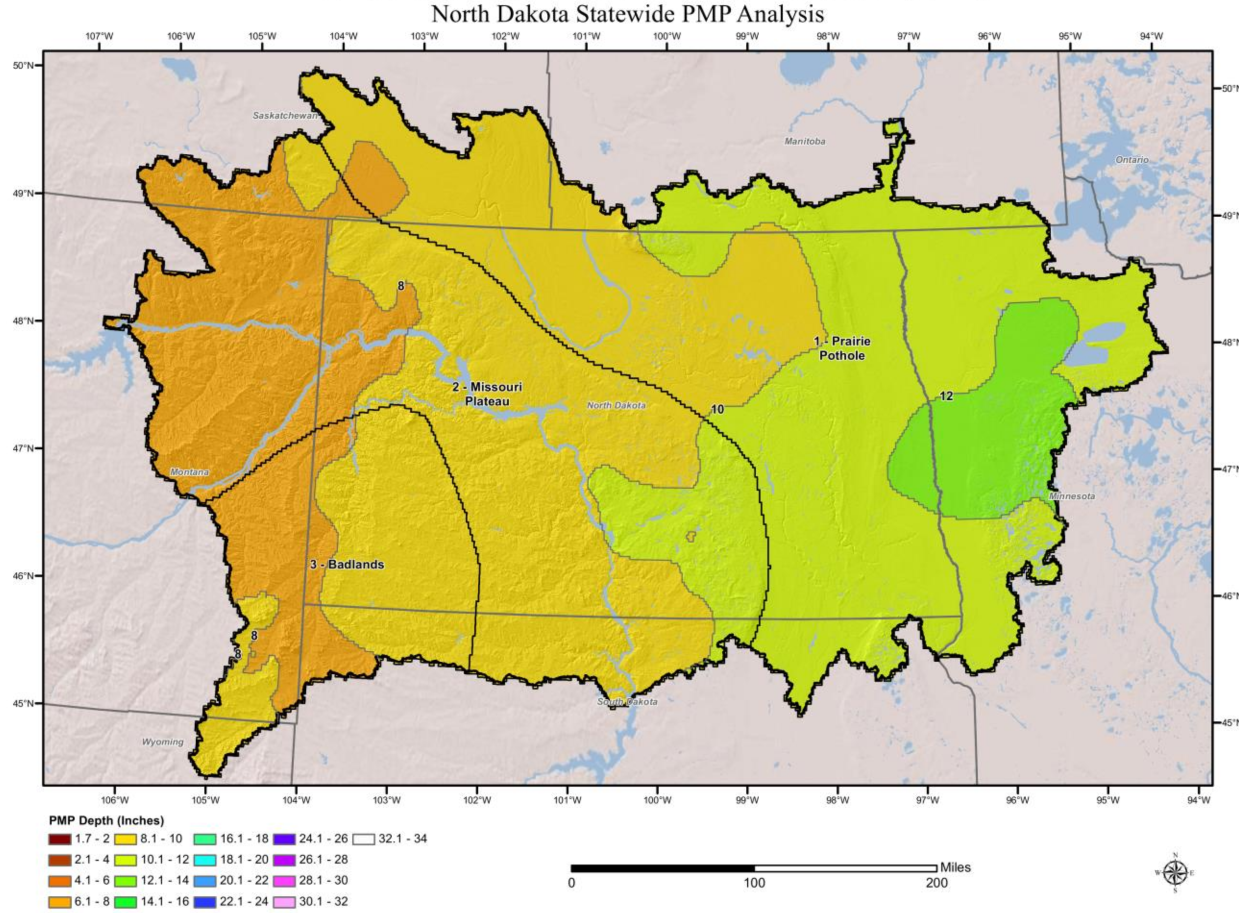
6-Hour Cool Season Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis



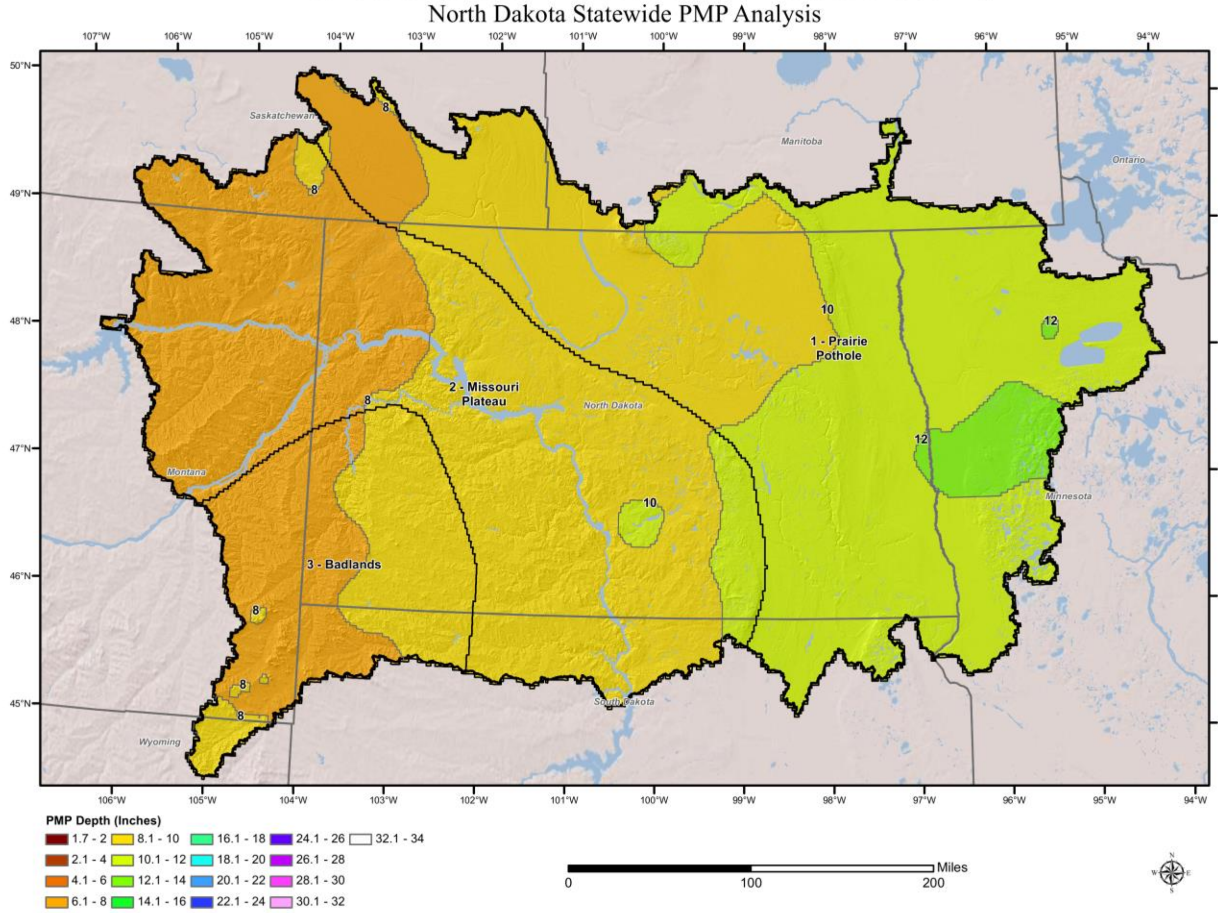
6-Hour Cool Season Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis



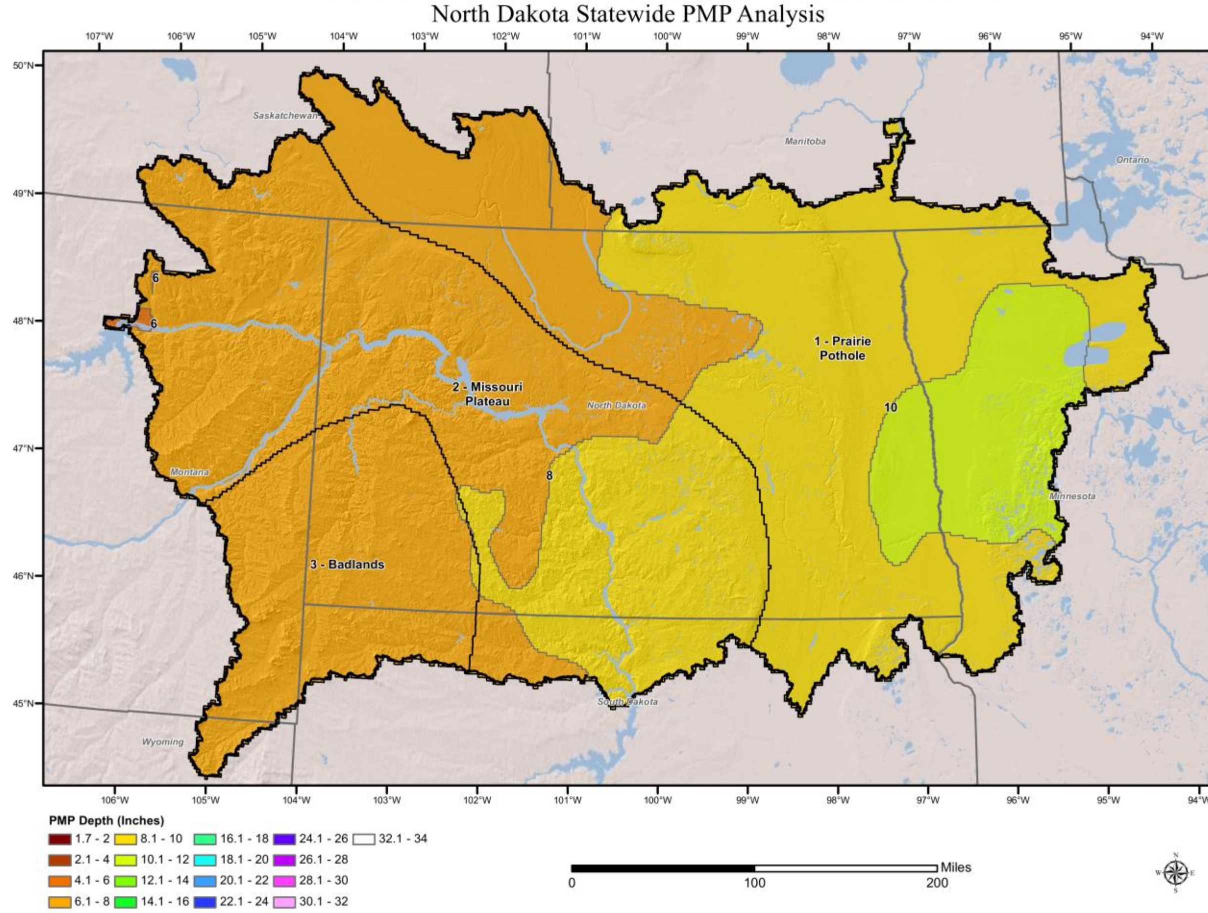
24-Hour Cool Season Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



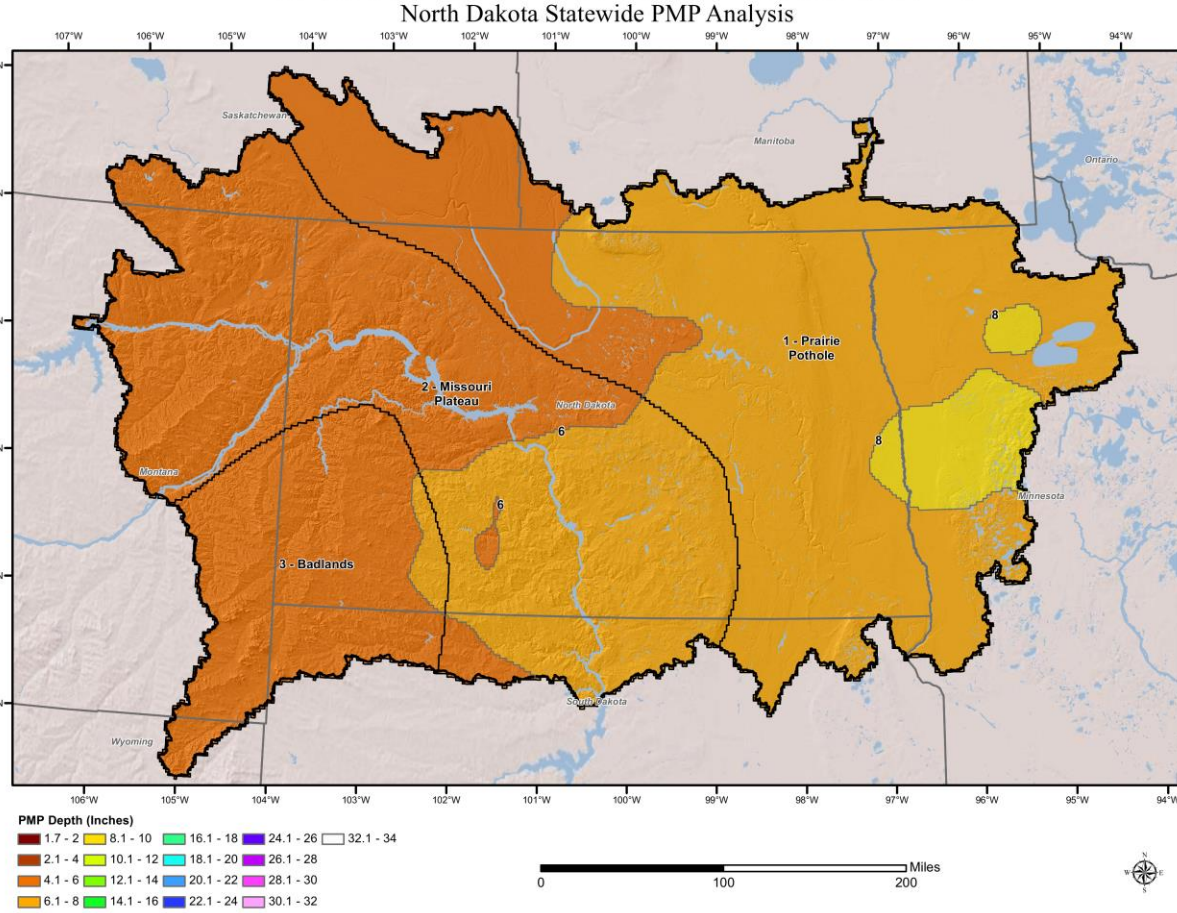
24-Hour Cool Season Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



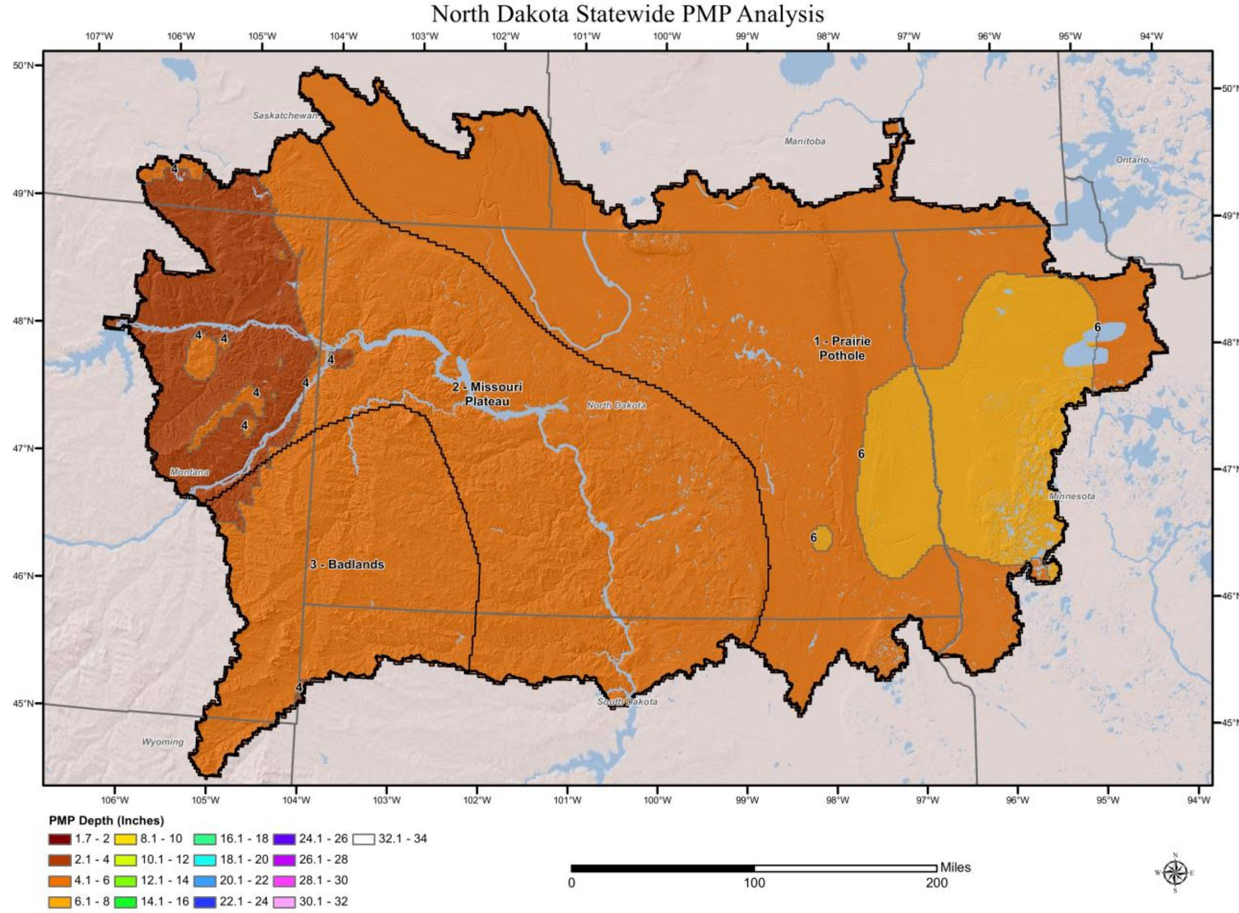
24-Hour Cool Season Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis



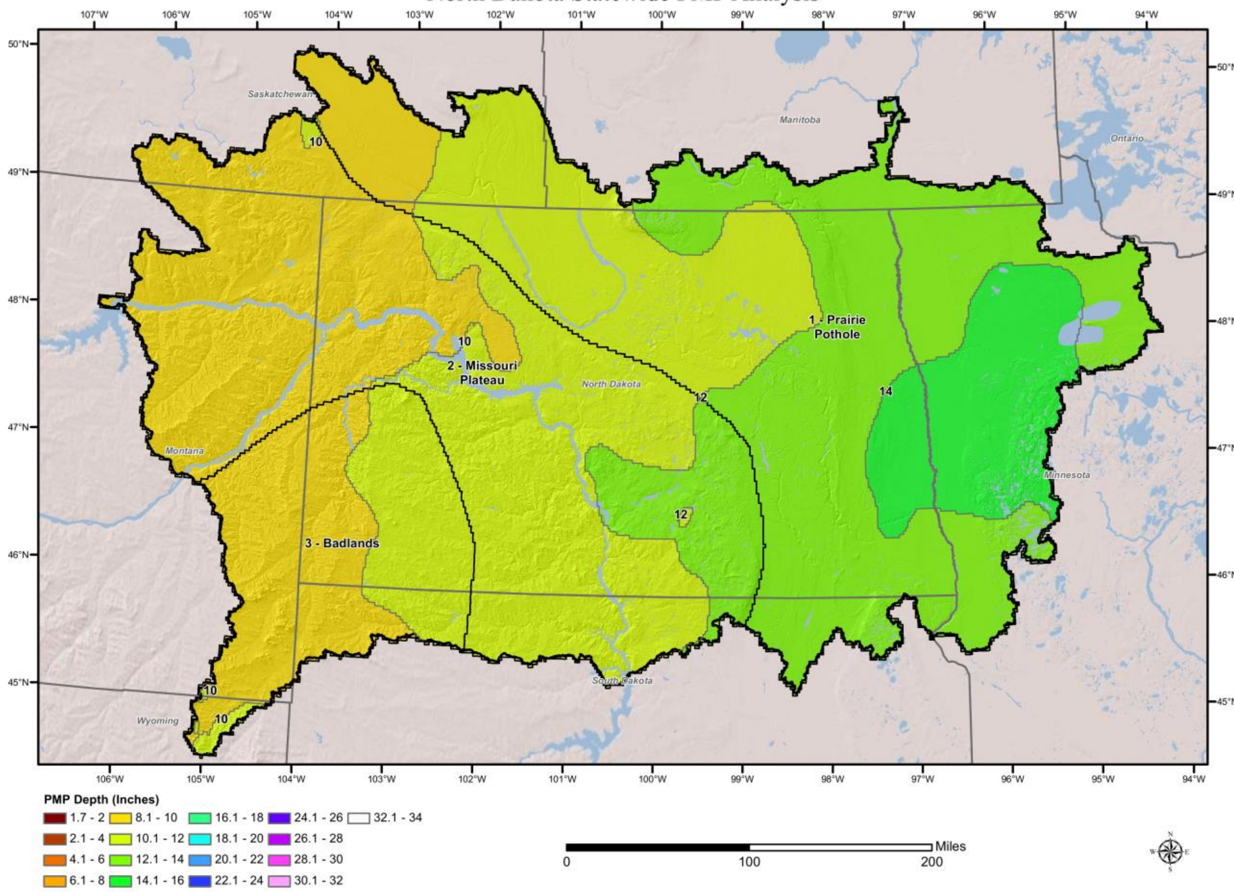
24-Hour Cool Season Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis



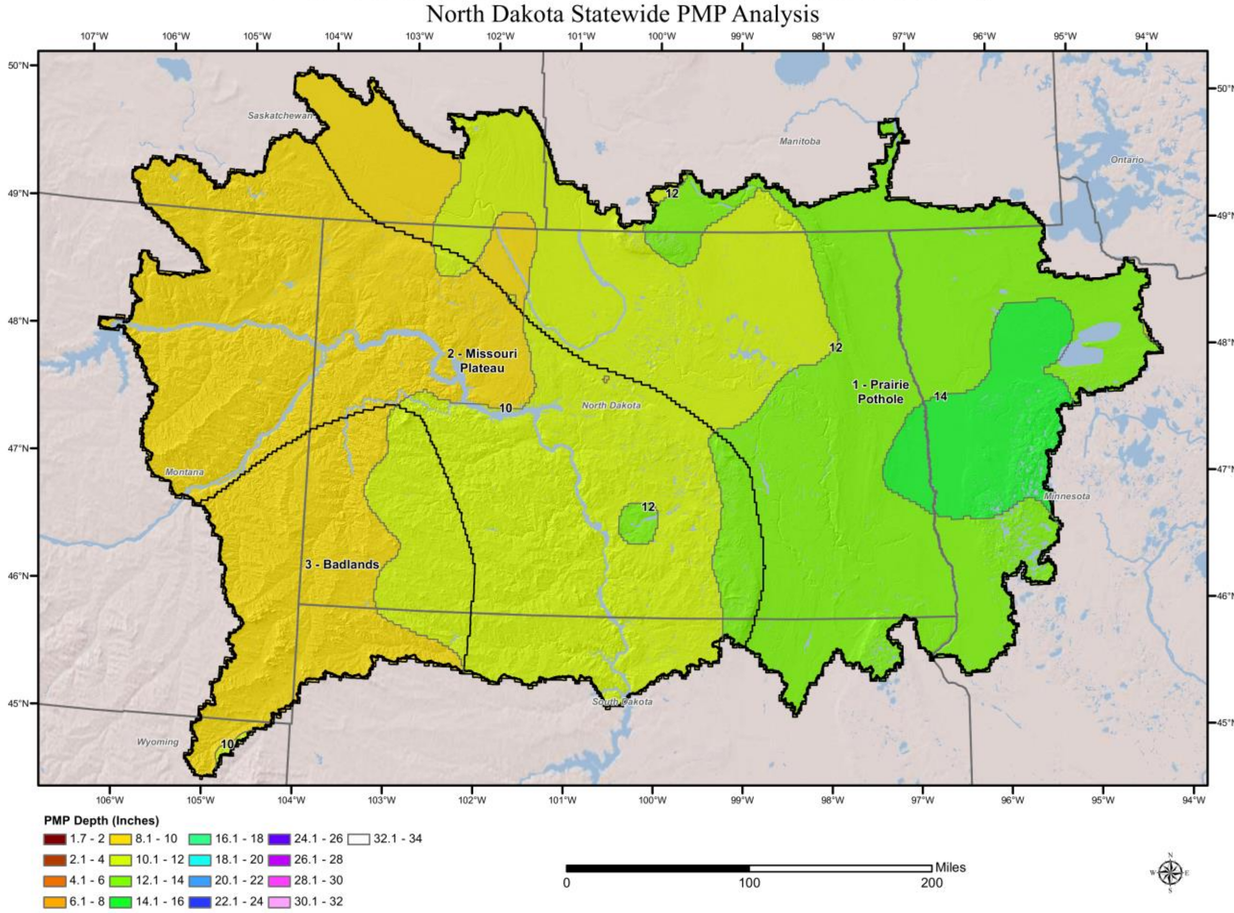
24-Hour Cool Season Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis



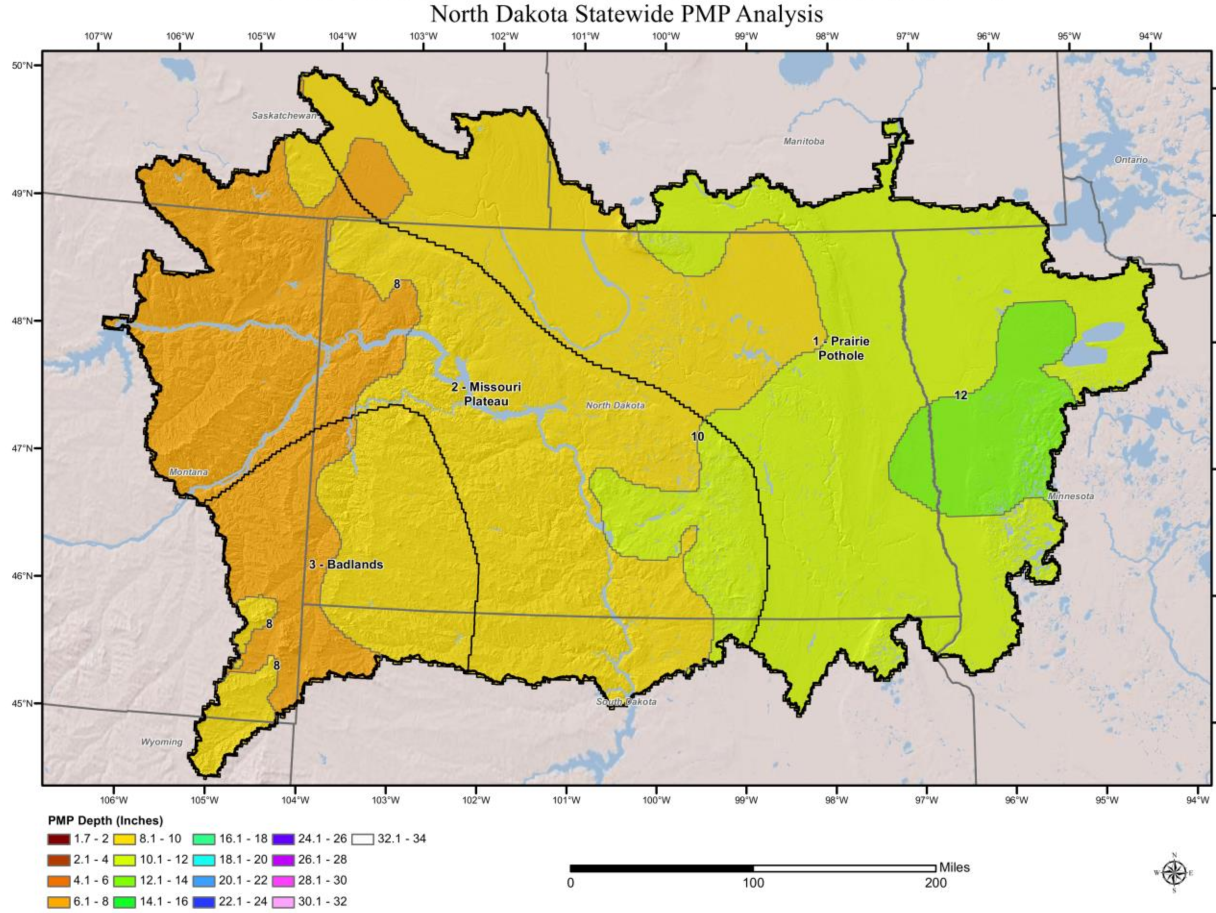
48-Hour Cool Season Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



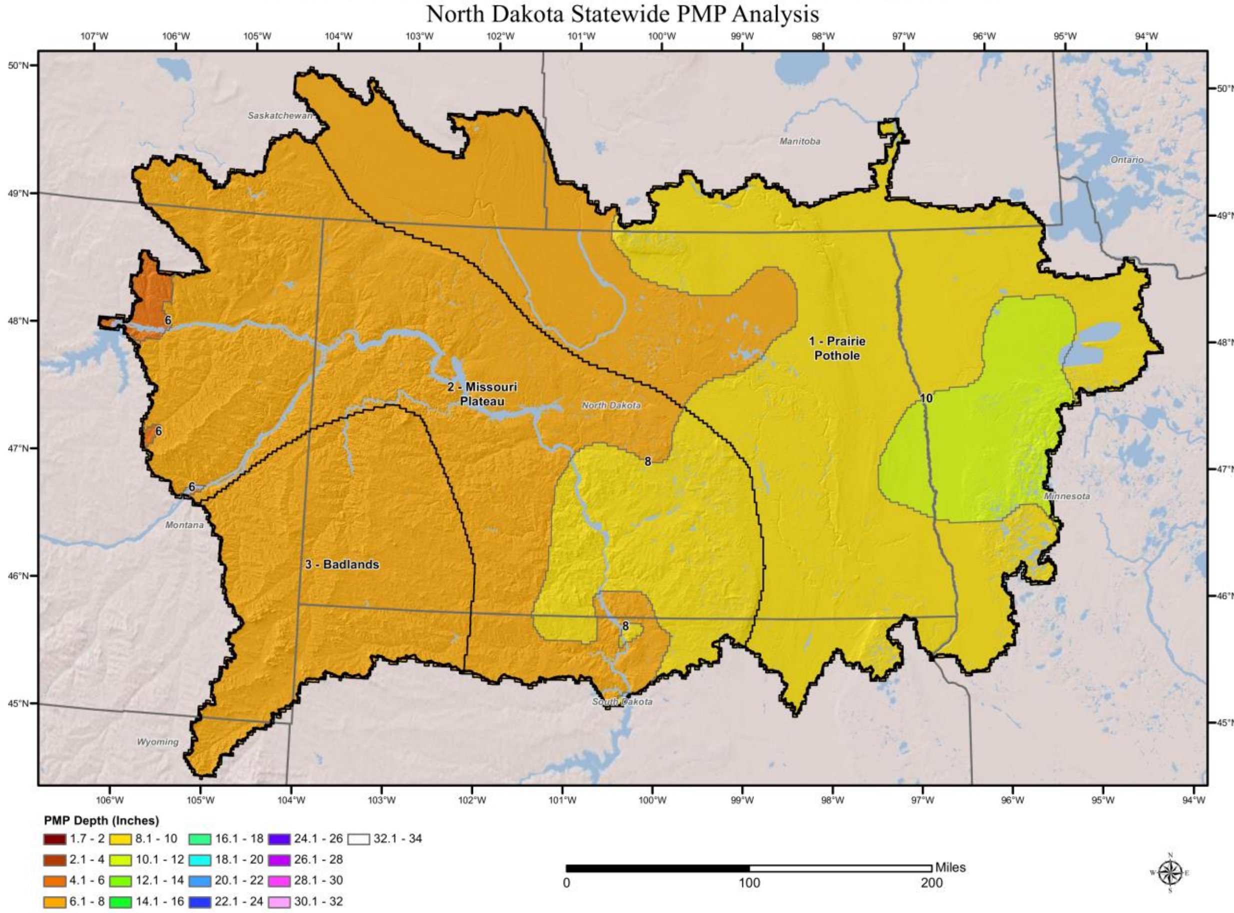
48-Hour Cool Season Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



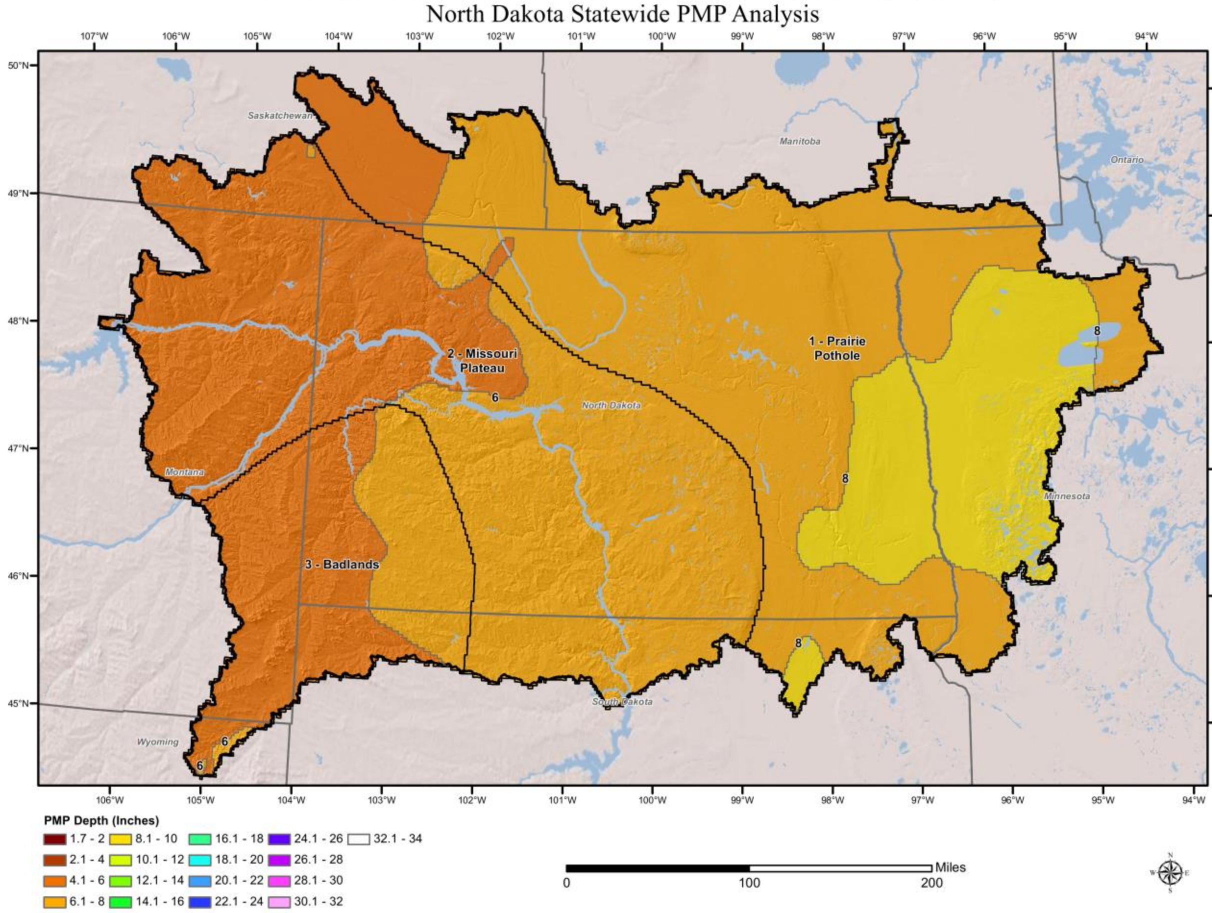
48-Hour Cool Season Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis



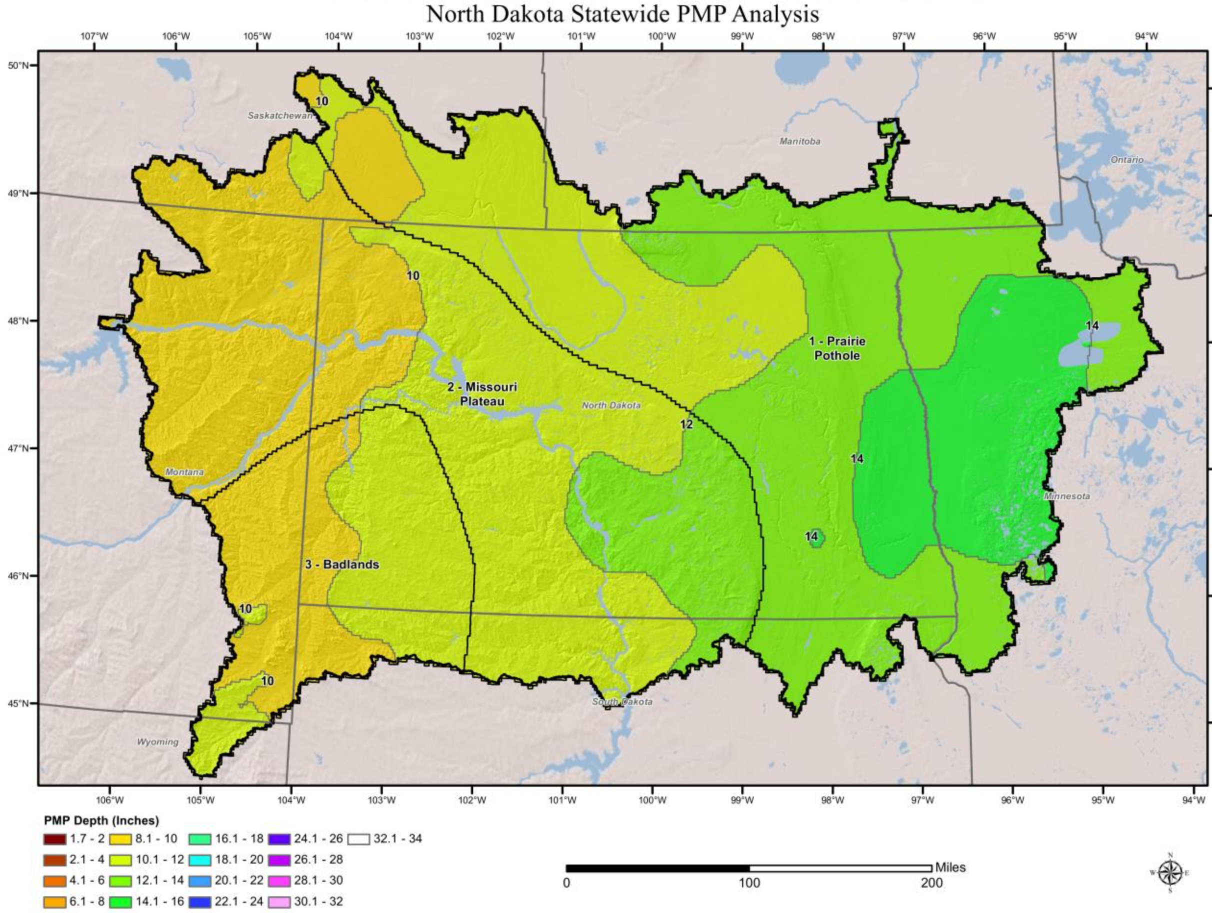
48-Hour Cool Season Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis



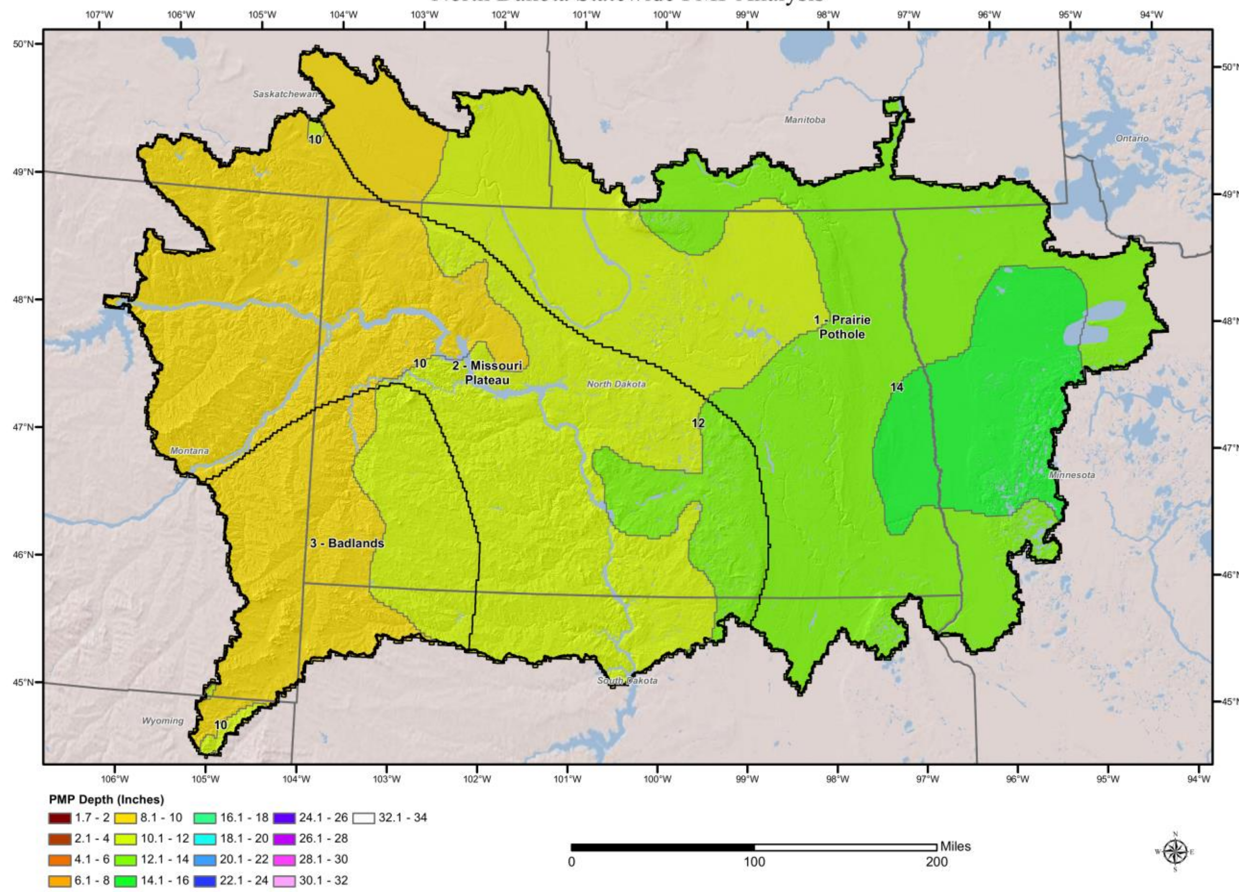
48-Hour Cool Season Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis



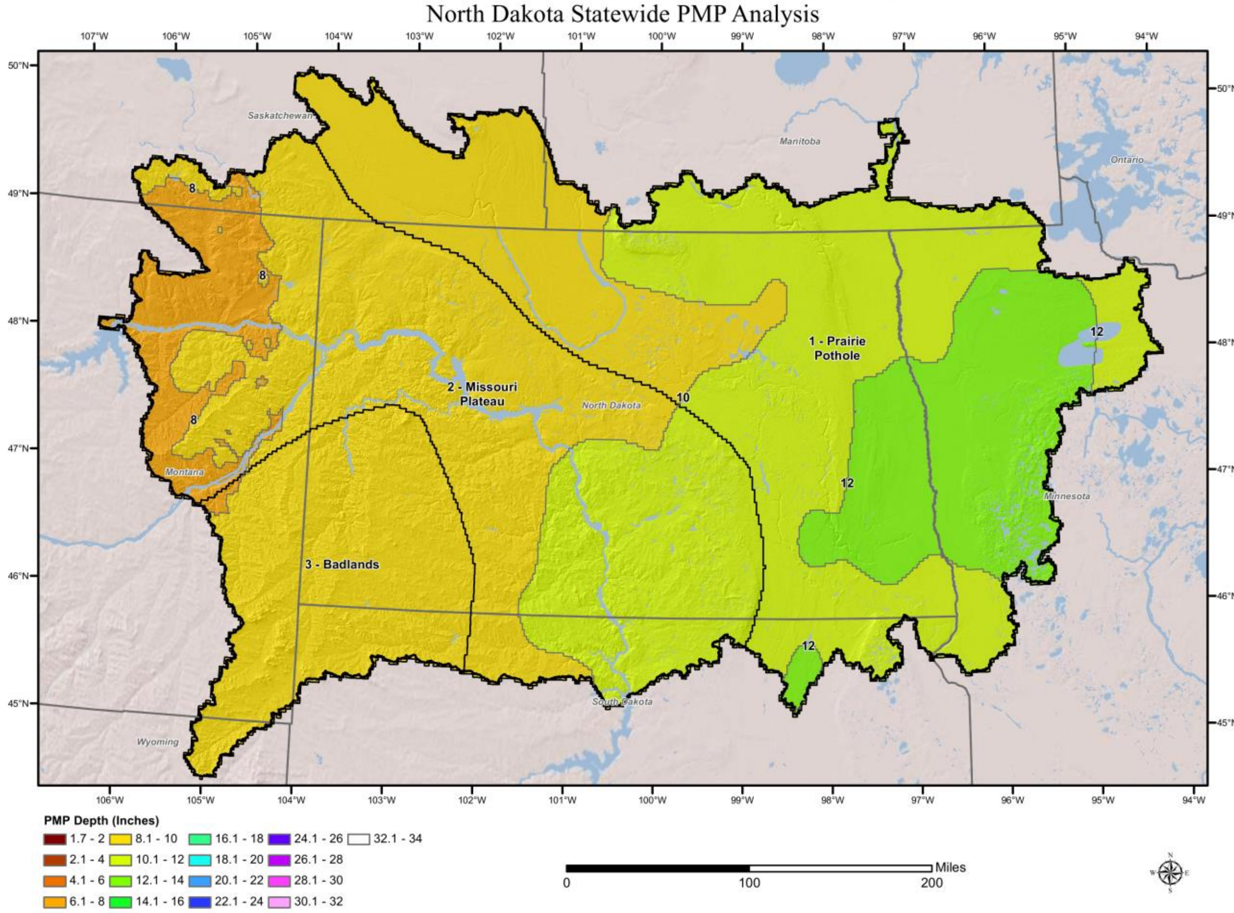
72-Hour Cool Season Storm Probable Maximum Precipitation (10 mi²)
North Dakota Statewide PMP Analysis



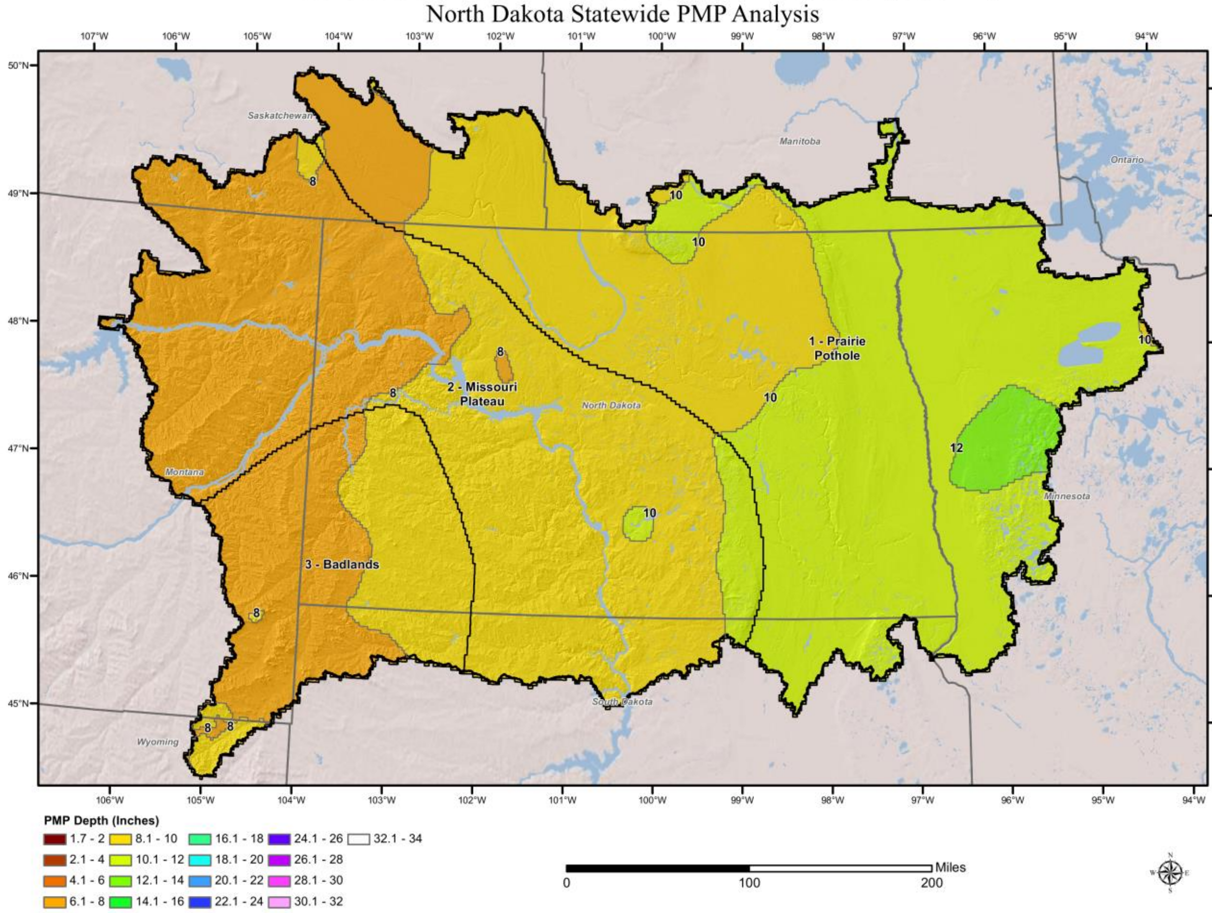
72-Hour Cool Season Storm Probable Maximum Precipitation (100 mi²)
North Dakota Statewide PMP Analysis



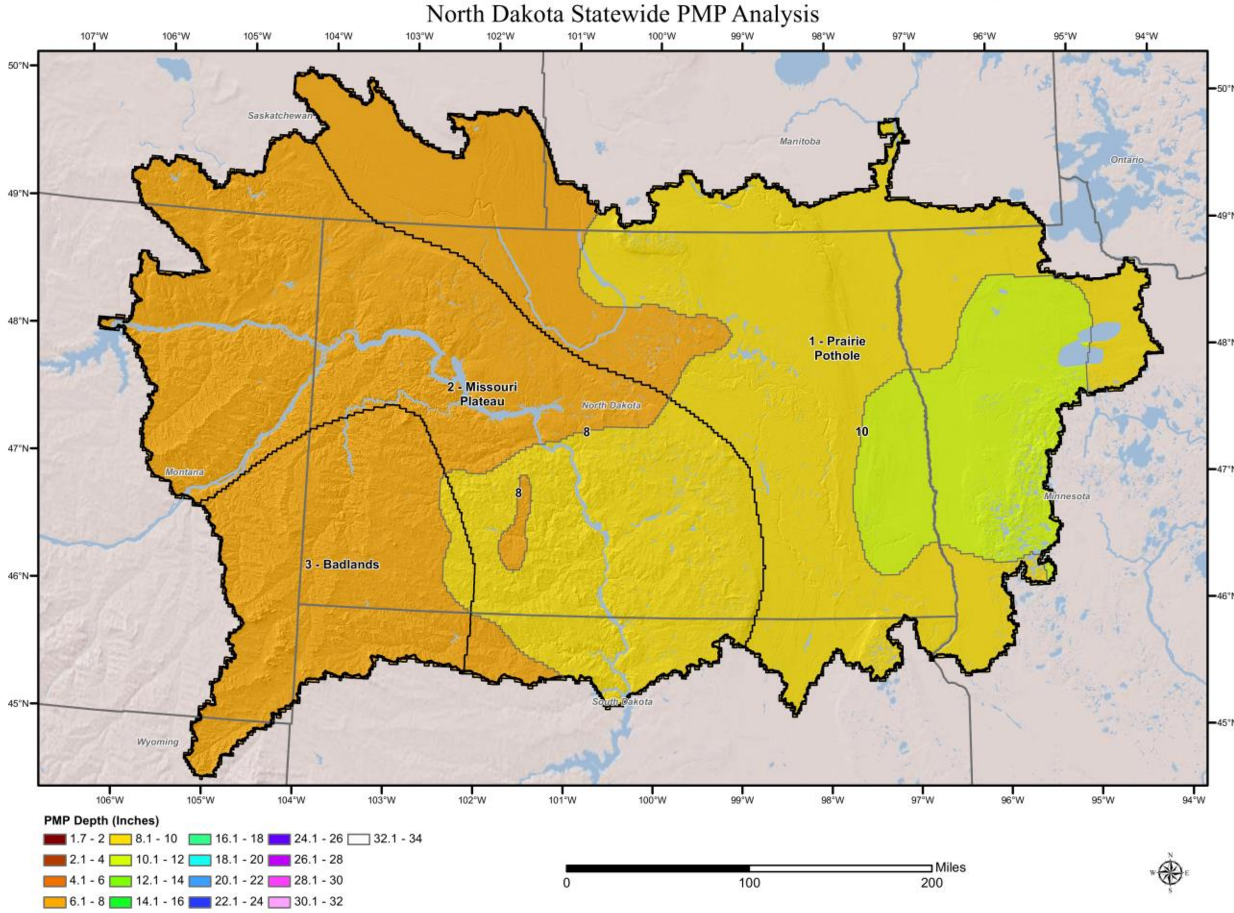
72-Hour Cool Season Storm Probable Maximum Precipitation (1000 mi²)
North Dakota Statewide PMP Analysis



72-Hour Cool Season Storm Probable Maximum Precipitation (5000 mi²)
North Dakota Statewide PMP Analysis



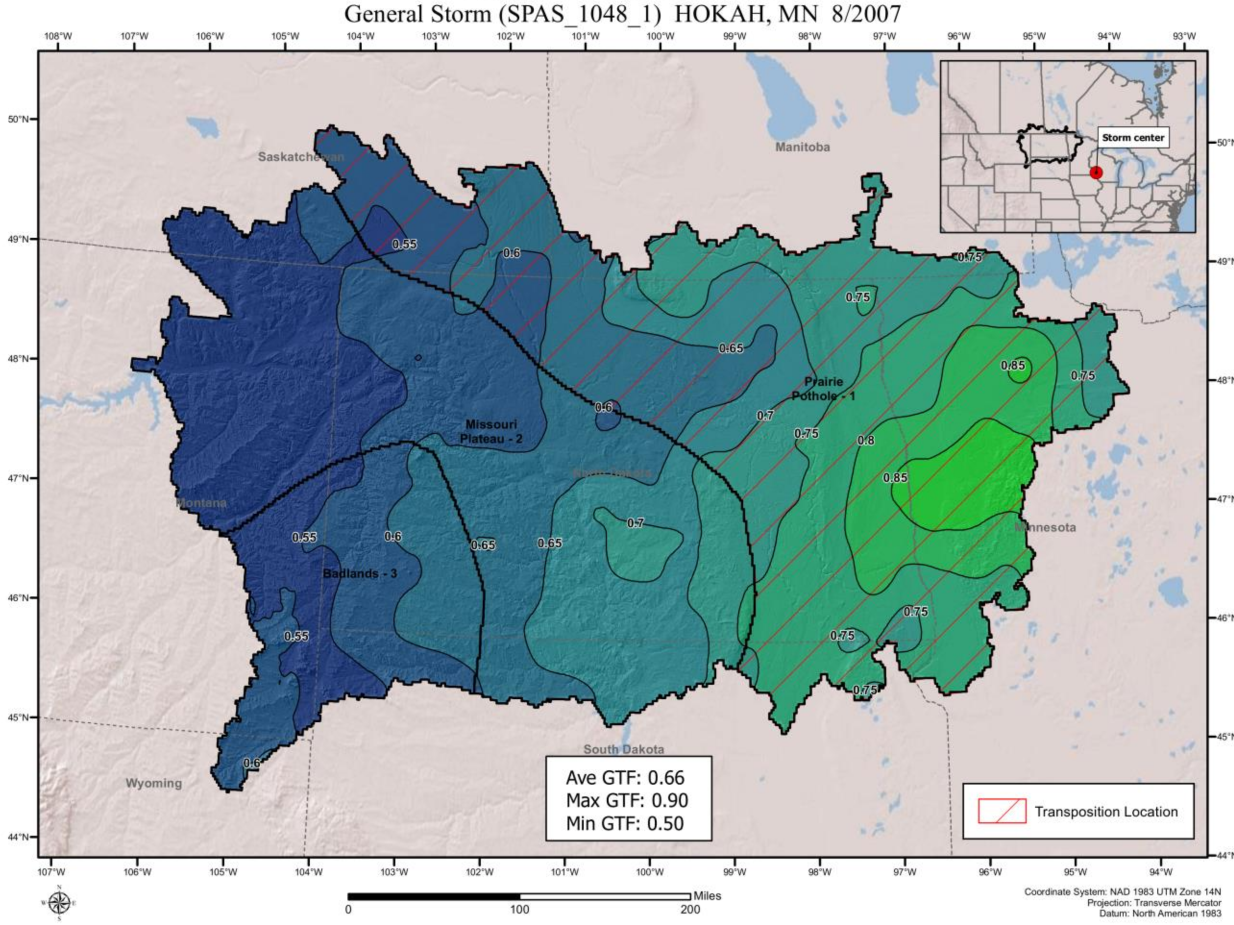
72-Hour Cool Season Storm Probable Maximum Precipitation (20000 mi²)
North Dakota Statewide PMP Analysis



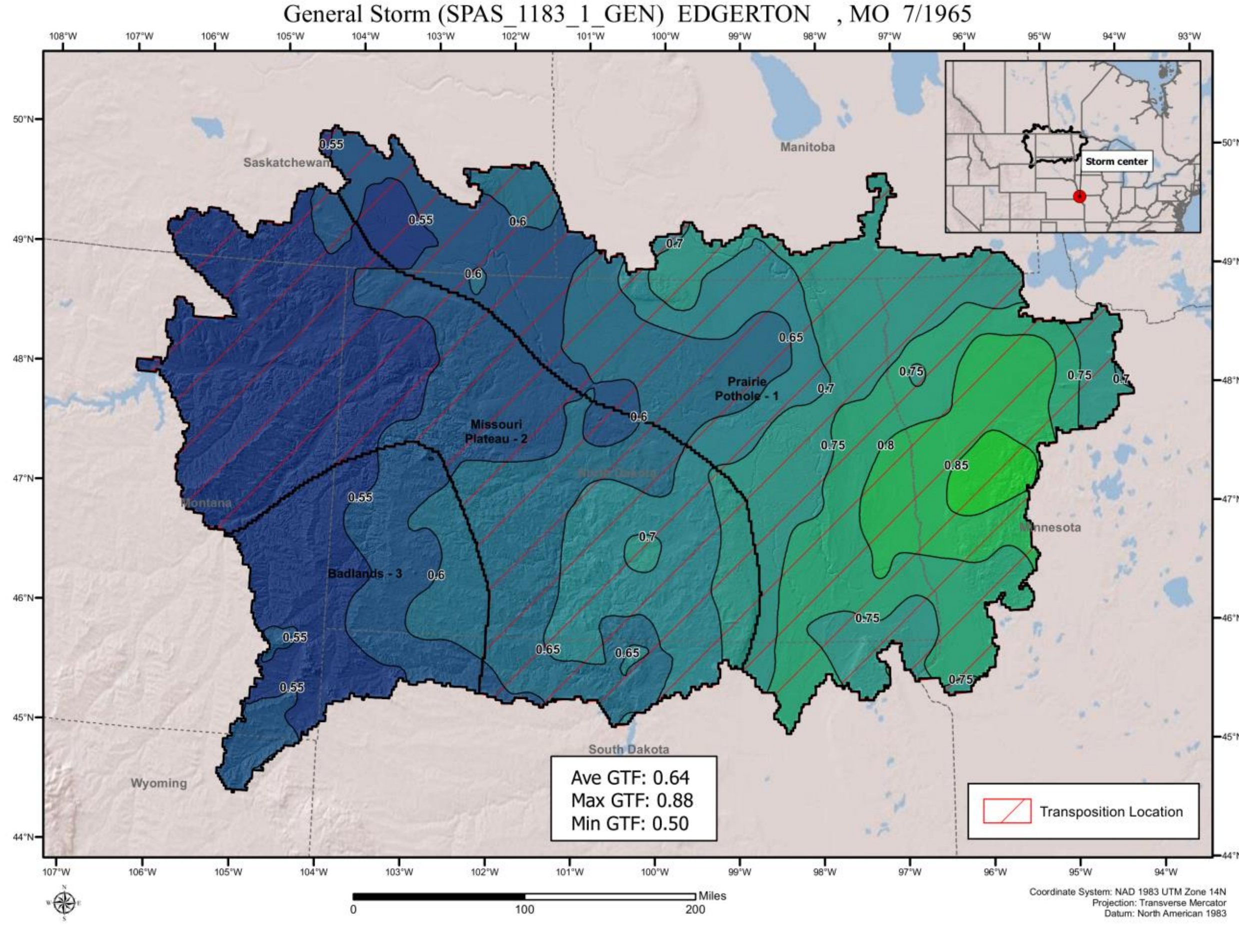
Appendix B
Geographic Transposition Factor (GTF) Maps

General Storms

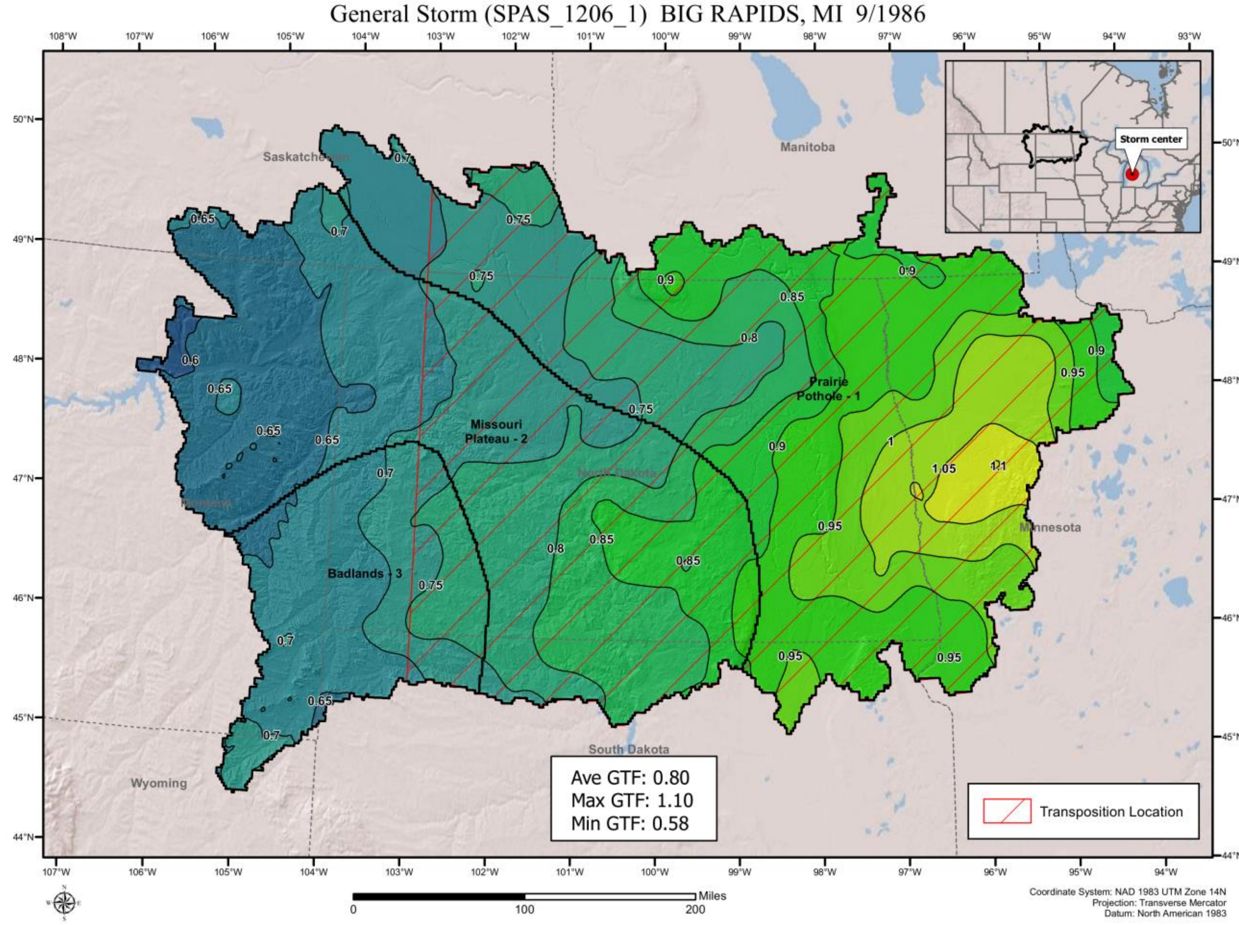
Geographic Transposition Factor
General Storm (SPAS_1048_1) HOKAH, MN 8/2007



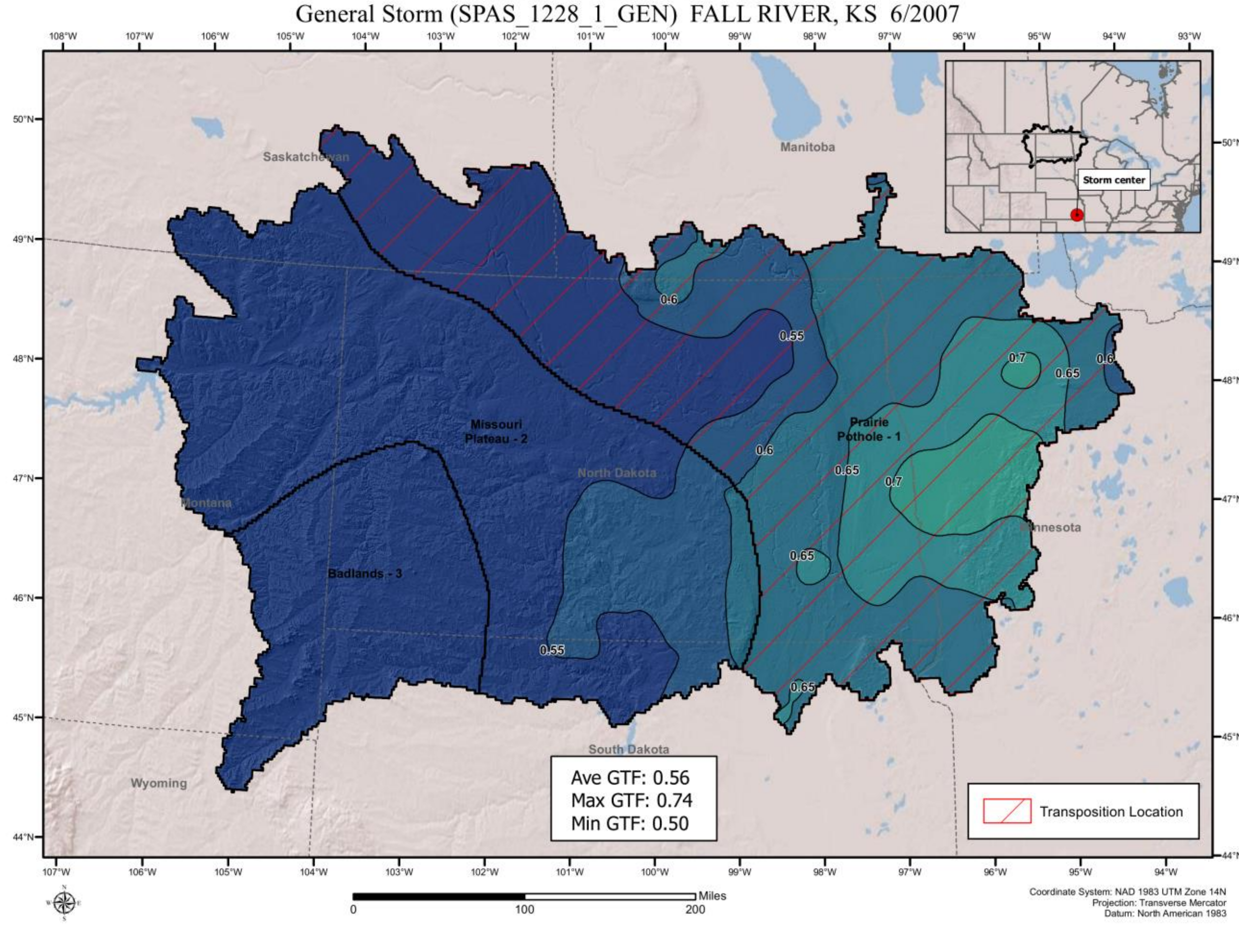
Geographic Transposition Factor
General Storm (SPAS_1183_1_GEN) EDGERTON , MO 7/1965



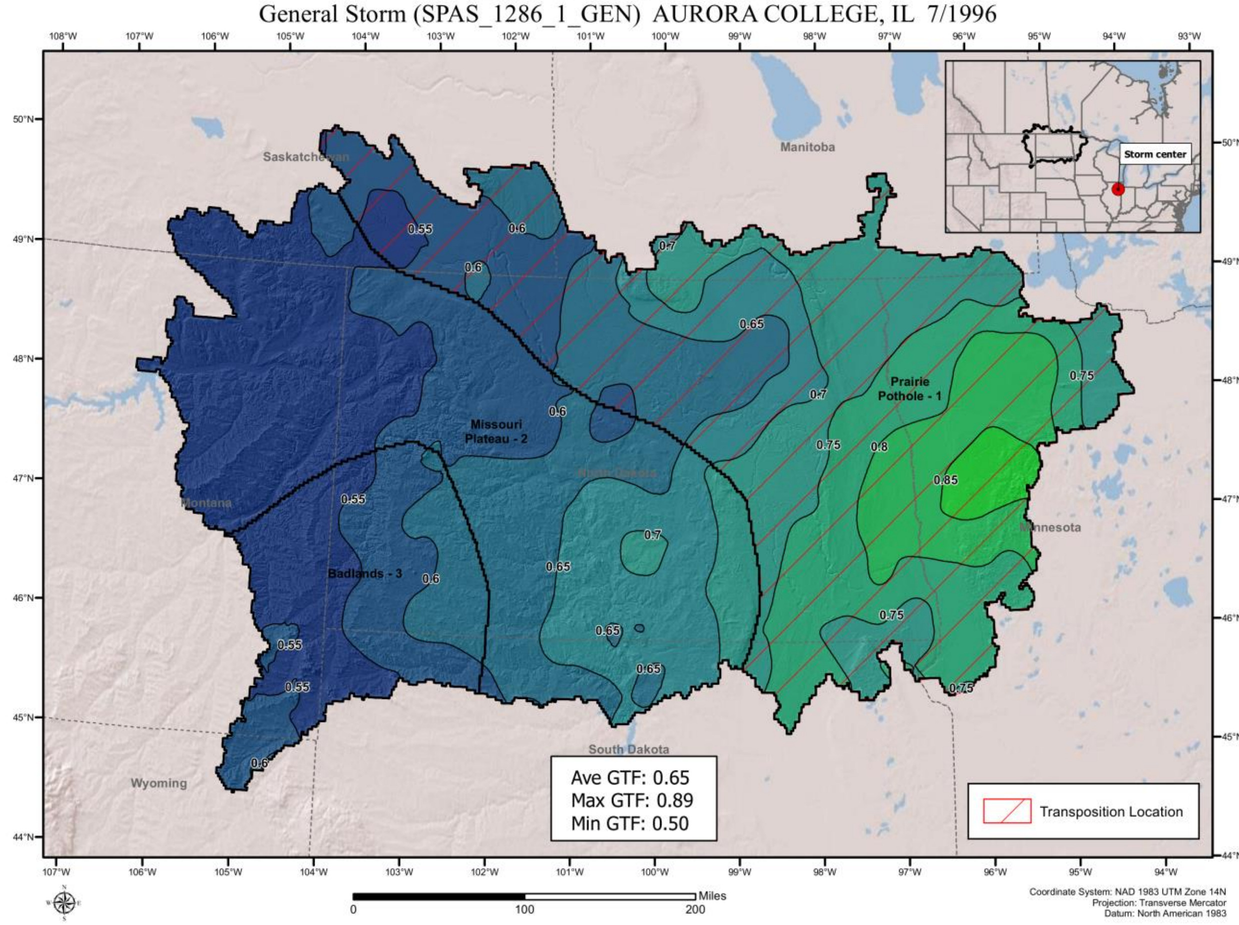
Geographic Transposition Factor
General Storm (SPAS_1206_1) BIG RAPIDS, MI 9/1986



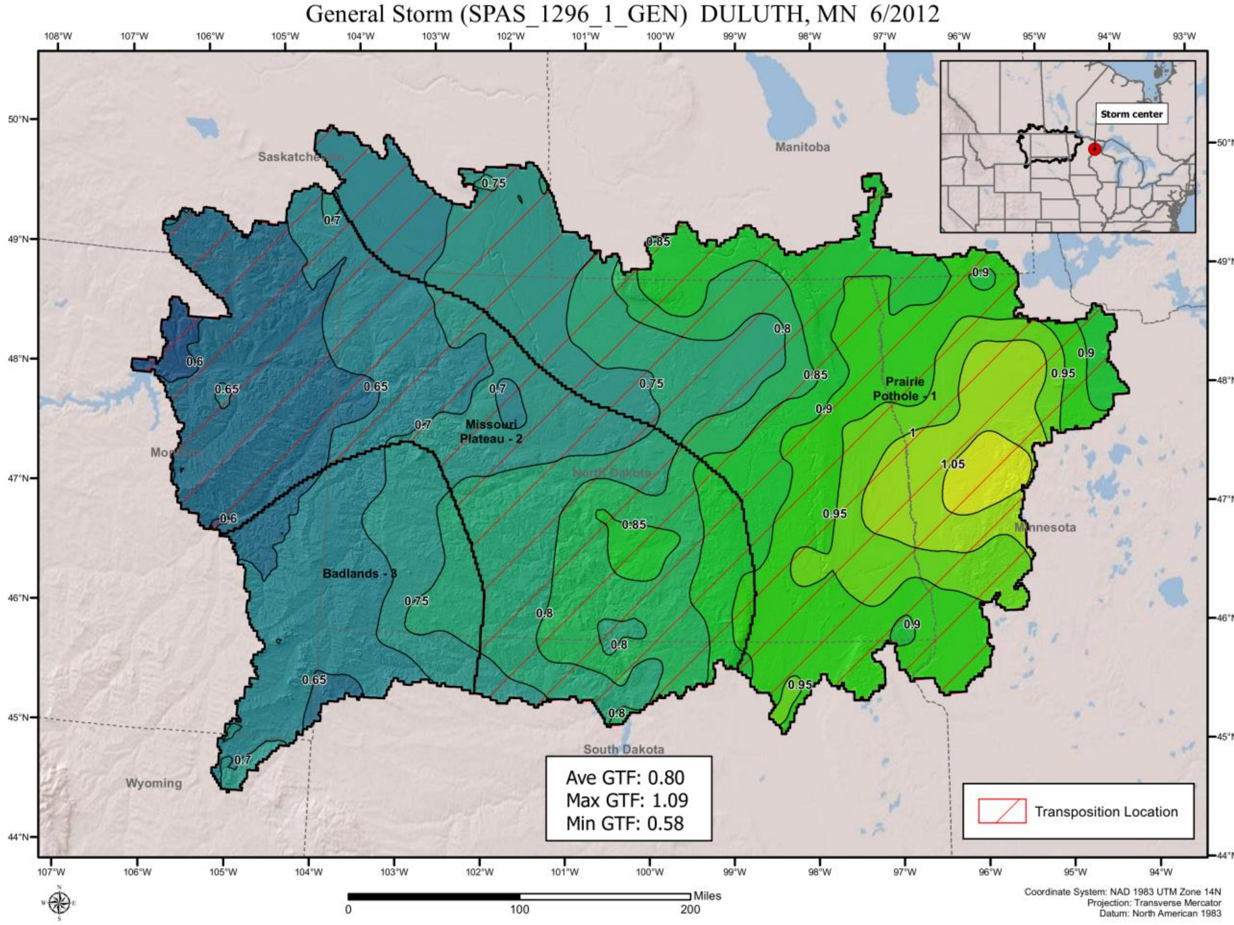
Geographic Transposition Factor
General Storm (SPAS_1228_1_GEN) FALL RIVER, KS 6/2007



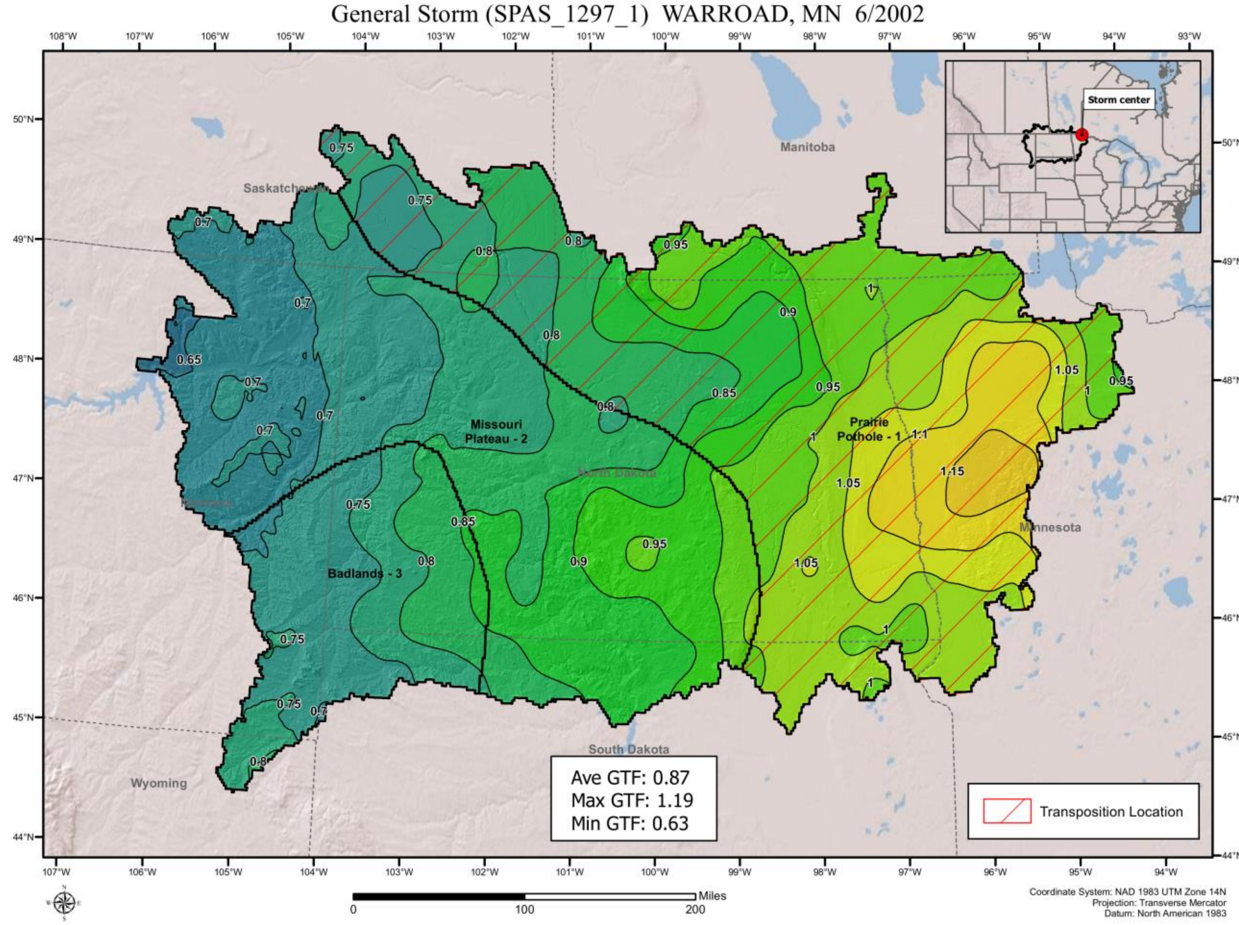
Geographic Transposition Factor
General Storm (SPAS_1286_1_GEN) AURORA COLLEGE, IL 7/1996



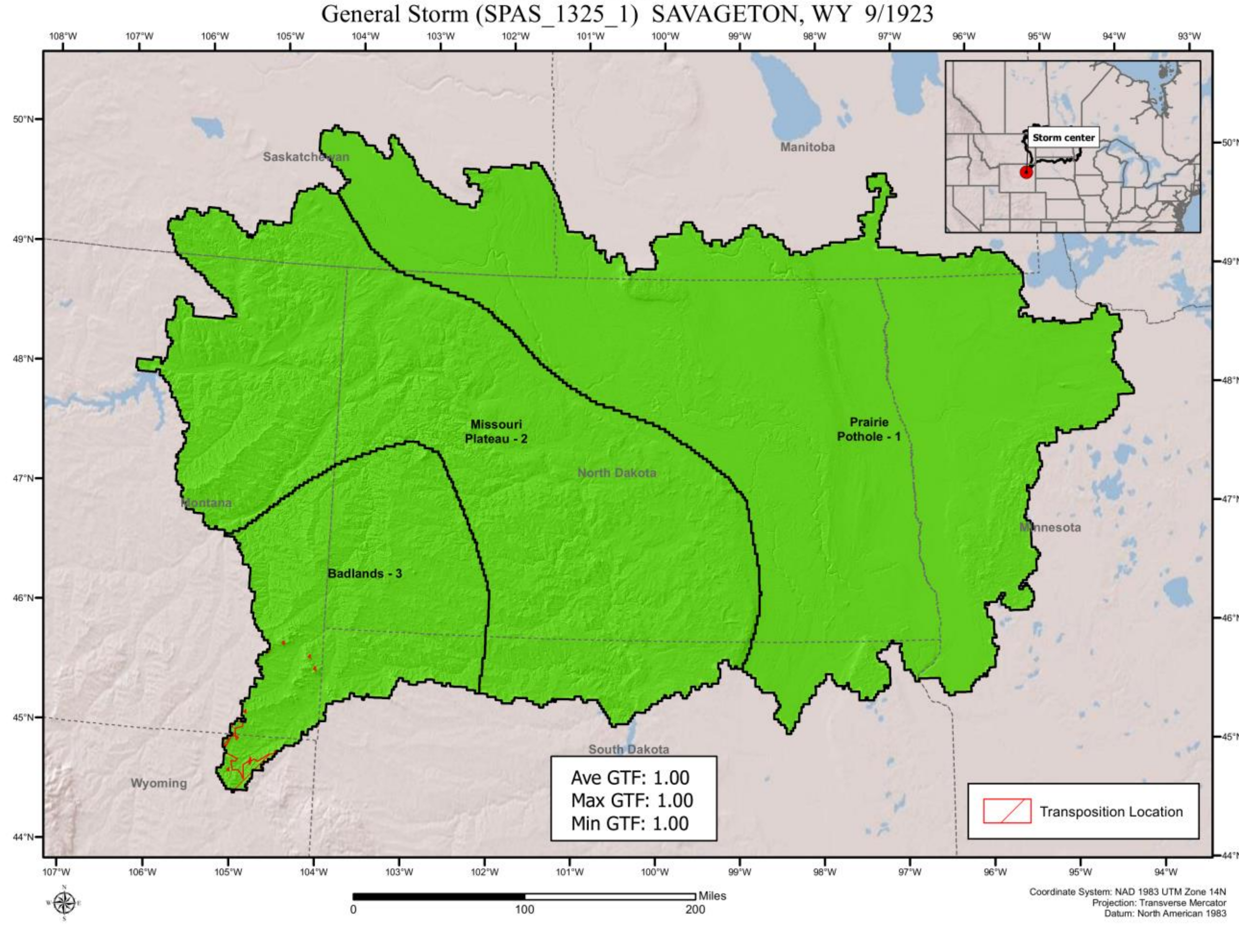
Geographic Transposition Factor
General Storm (SPAS_1296_1_GEN) DULUTH, MN 6/2012



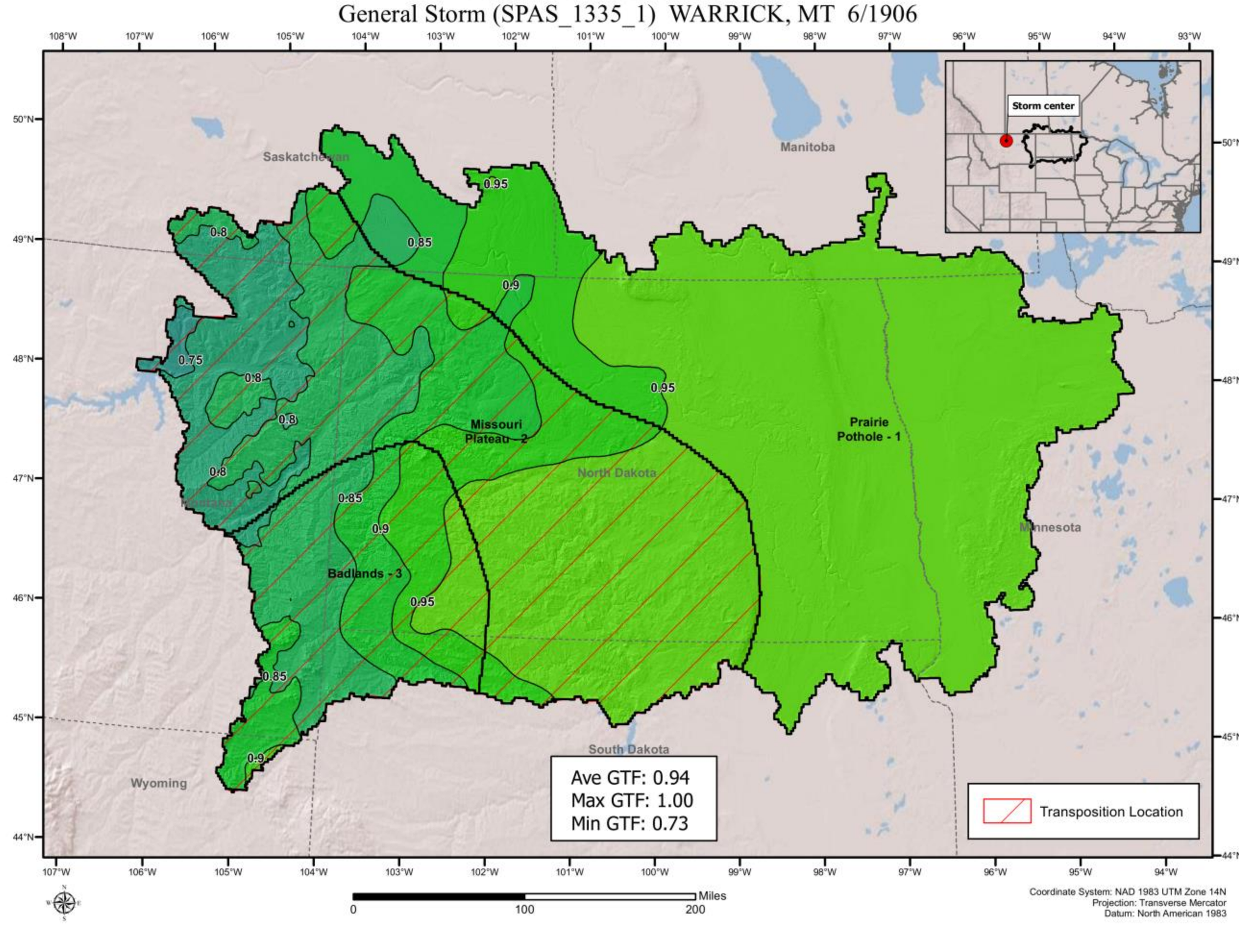
Geographic Transposition Factor
General Storm (SPAS_1297_1) WARROAD, MN 6/2002



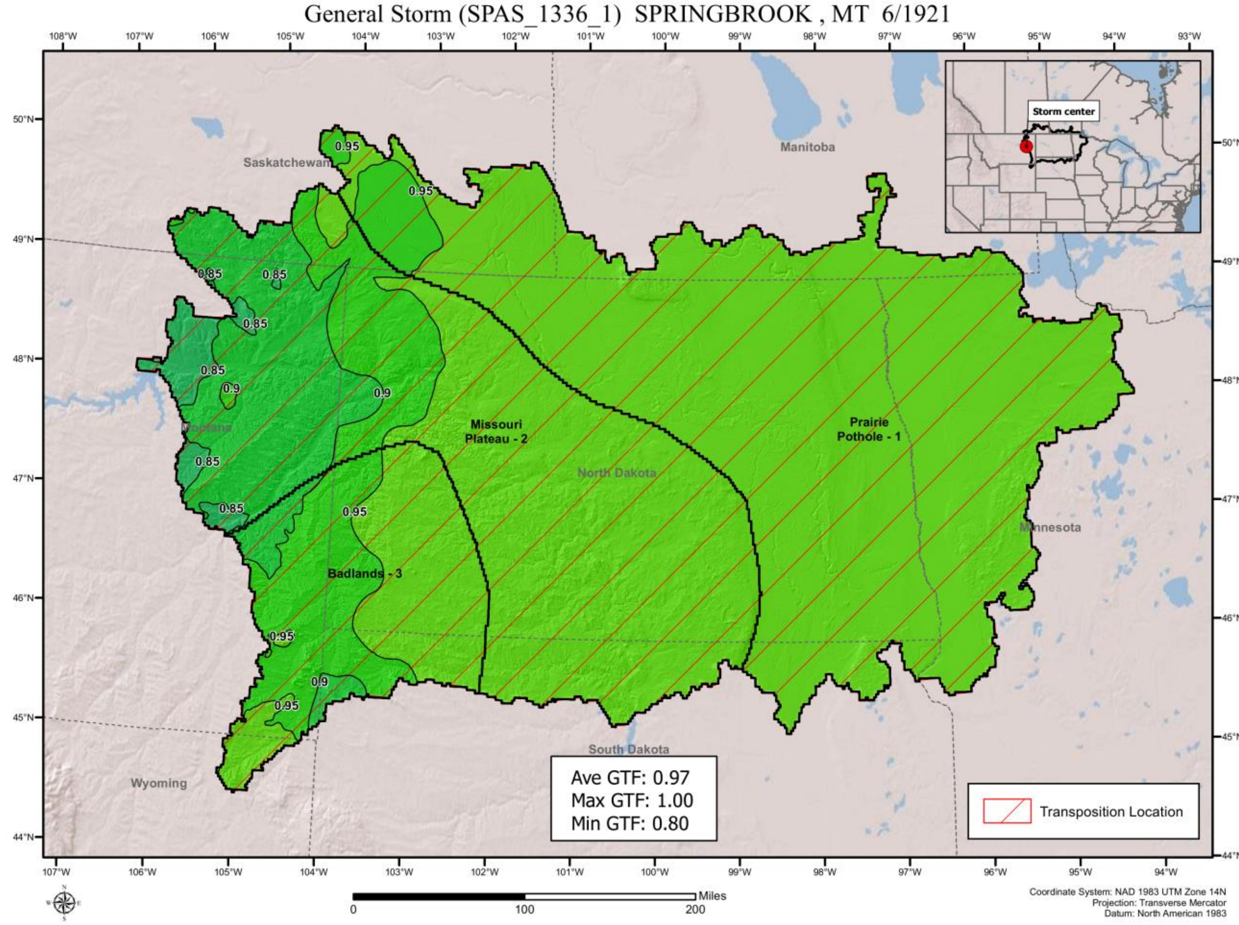
Geographic Transposition Factor
General Storm (SPAS_1325_1) SAVAGETON, WY 9/1923



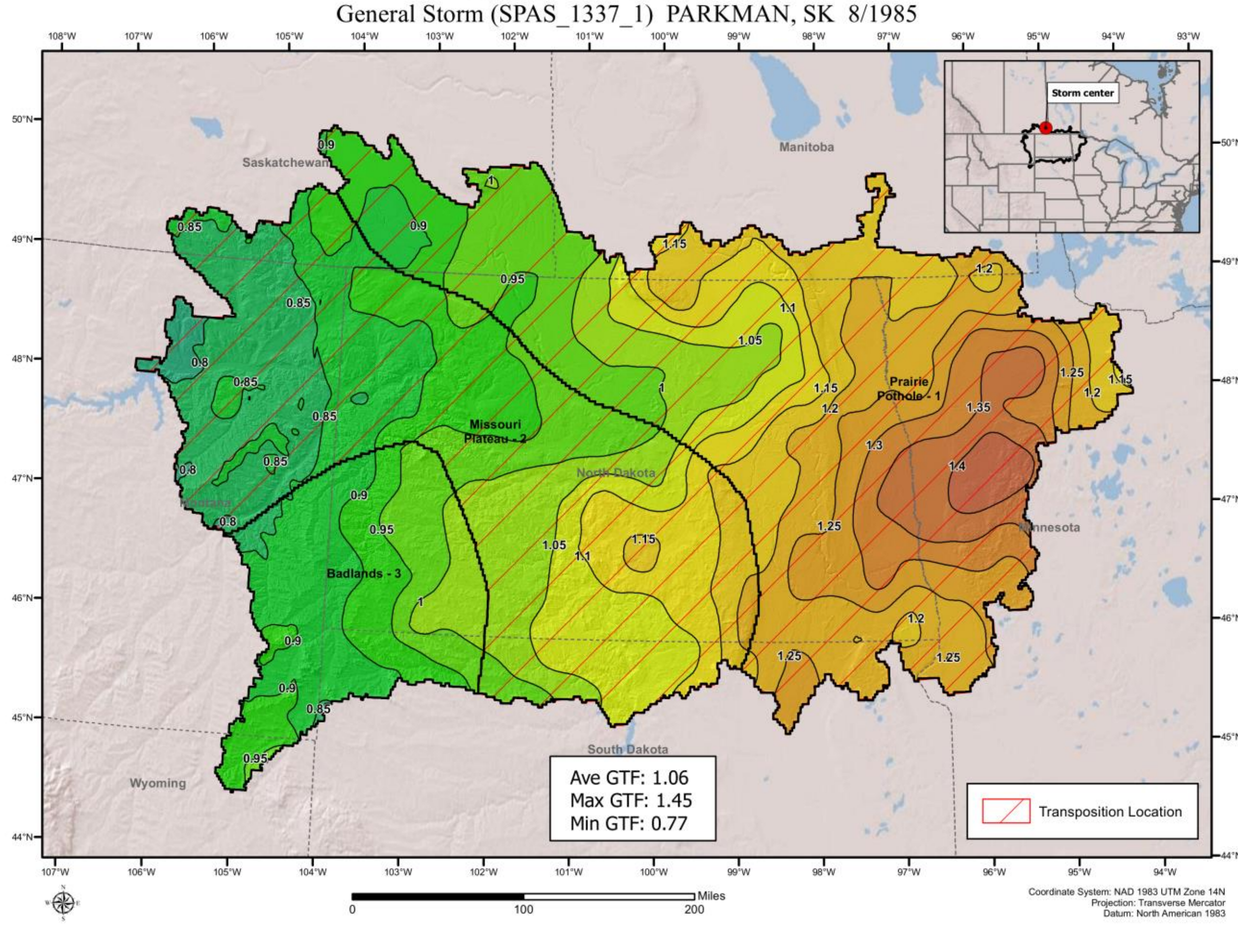
Geographic Transposition Factor
General Storm (SPAS_1335_1) WARRICK, MT 6/1906



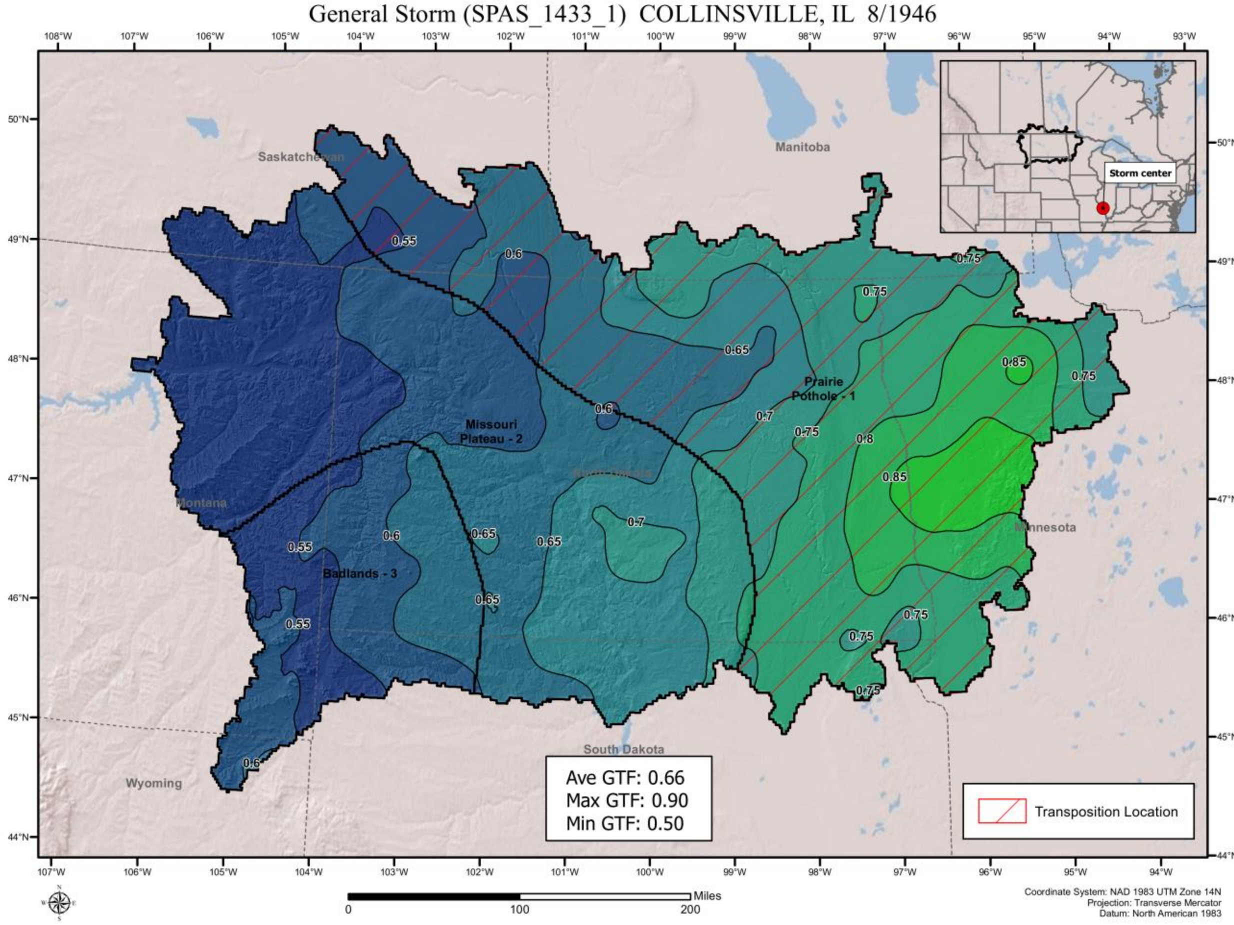
Geographic Transposition Factor
General Storm (SPAS_1336_1) SPRINGBROOK , MT 6/1921



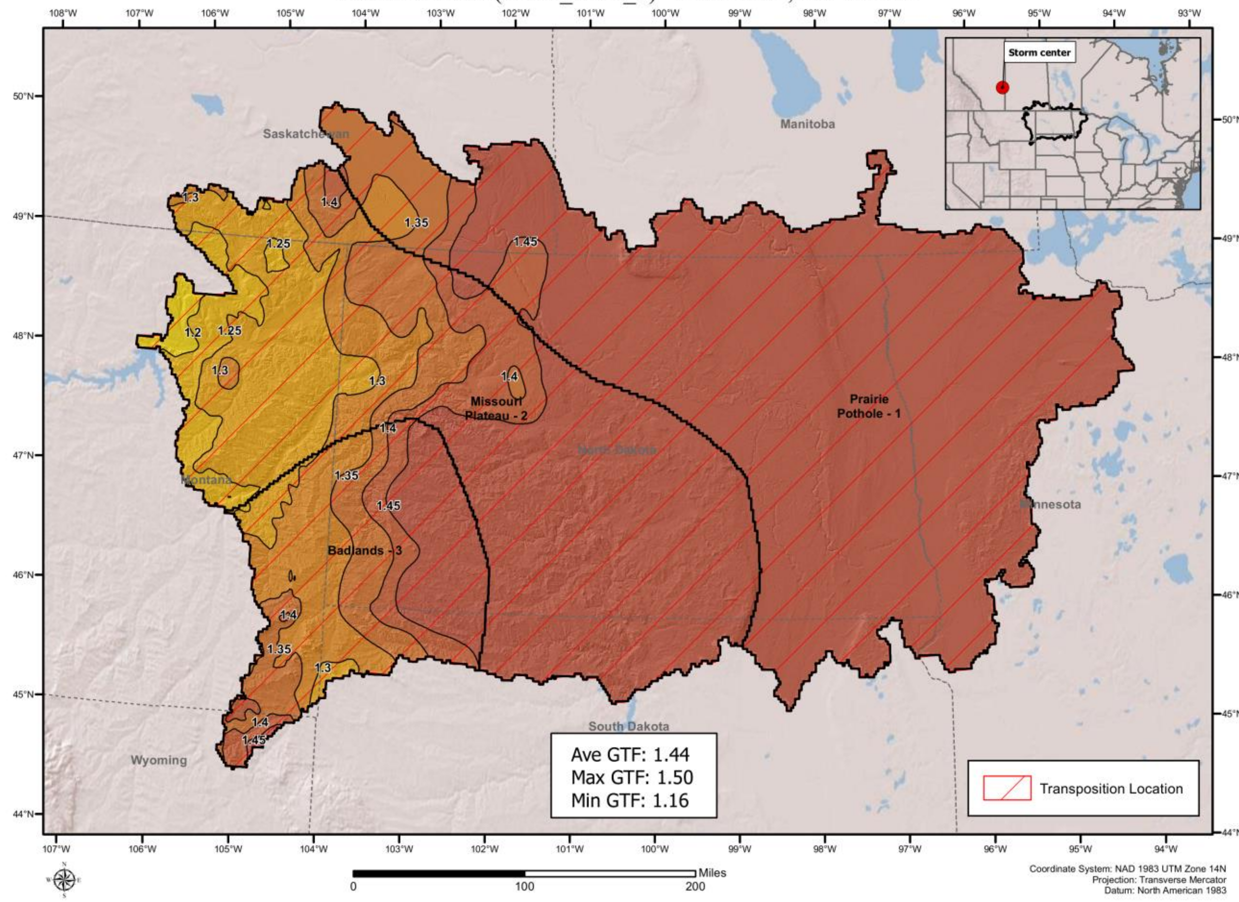
Geographic Transposition Factor
General Storm (SPAS_1337_1) PARKMAN, SK 8/1985



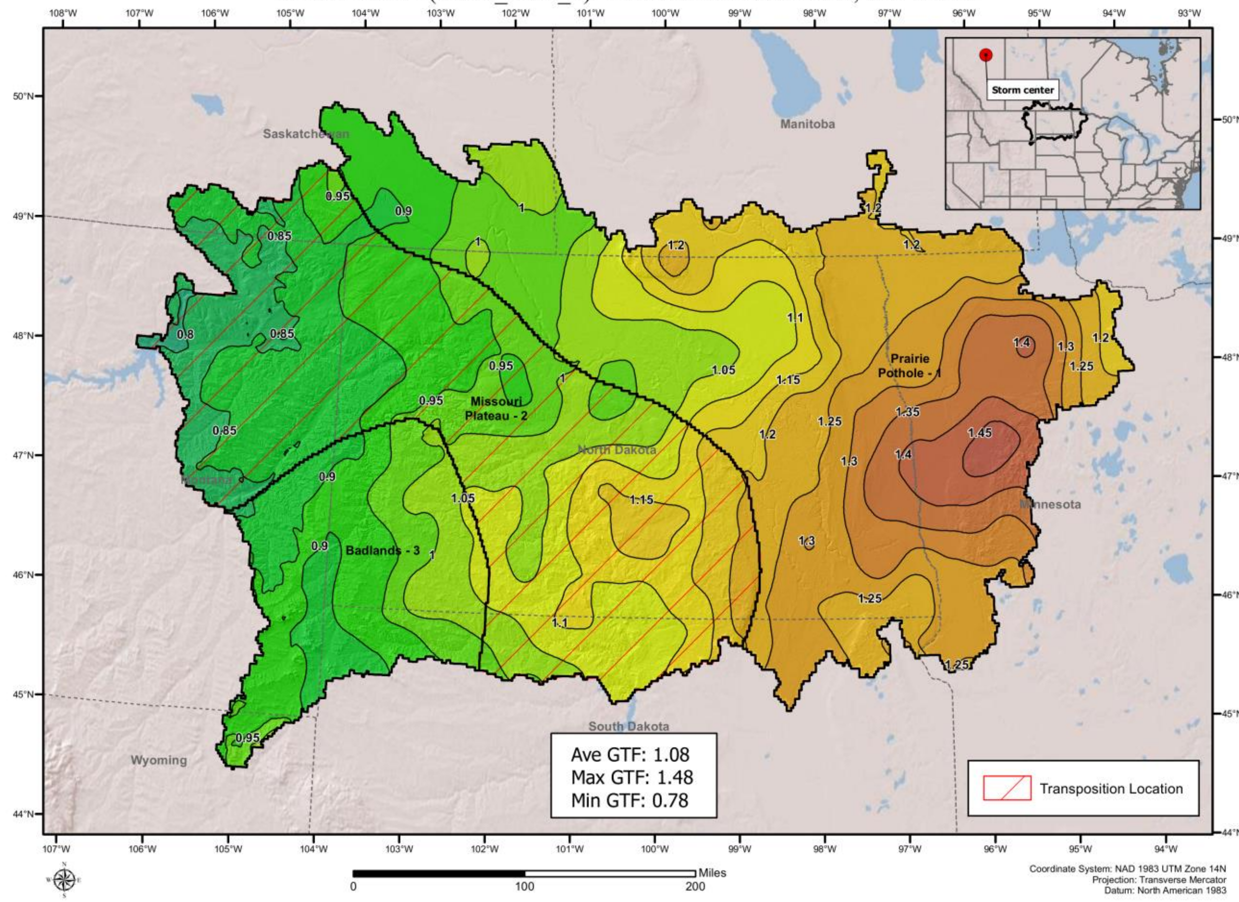
Geographic Transposition Factor
General Storm (SPAS_1433_1) COLLINSVILLE, IL 8/1946



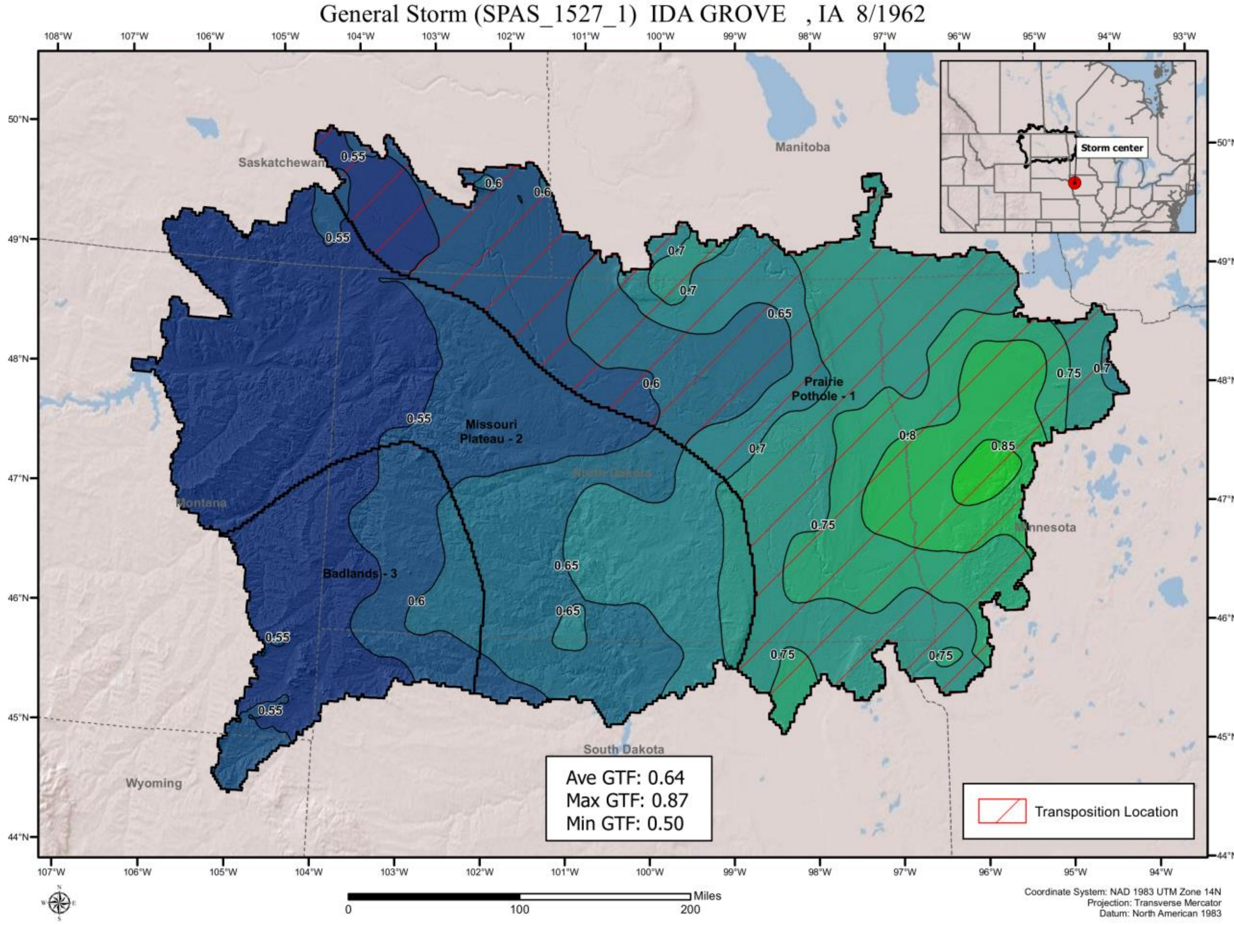
Geographic Transposition Factor
General Storm (SPAS_1502_1) VETERAN, AB 6/1973



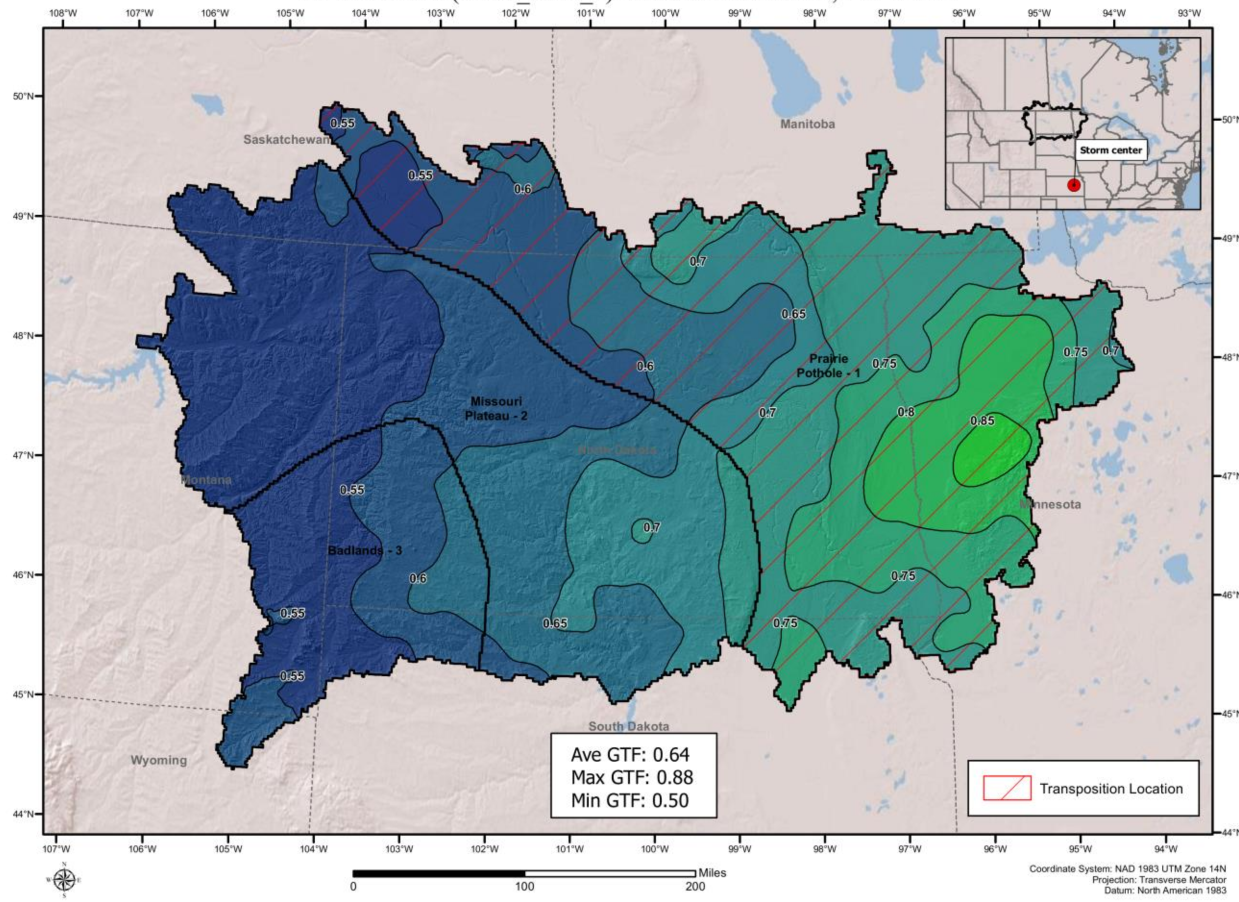
Geographic Transposition Factor
General Storm (SPAS_1504_1) PELICAN MOUNTAIN, AB 6/1970



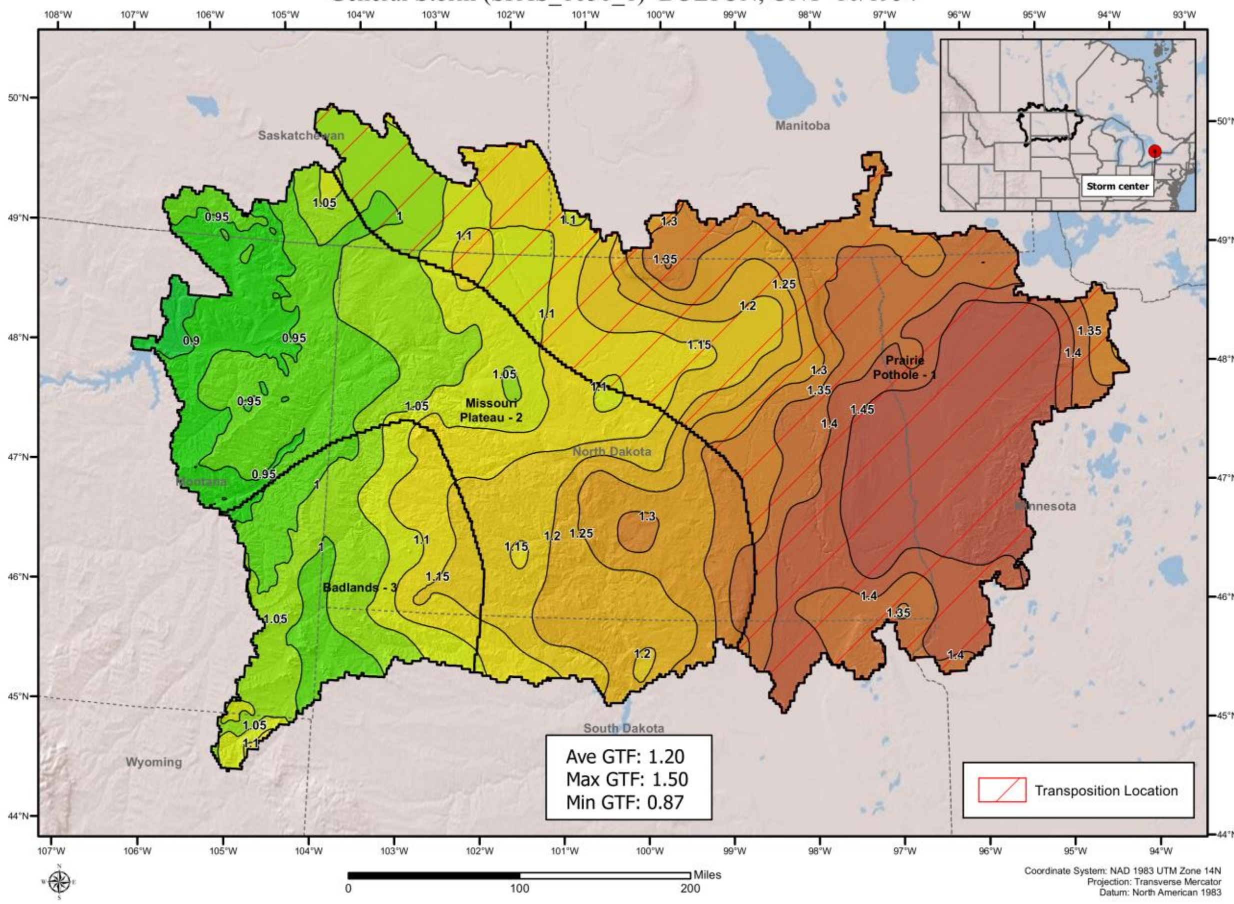
Geographic Transposition Factor
General Storm (SPAS_1527_1) IDA GROVE , IA 8/1962

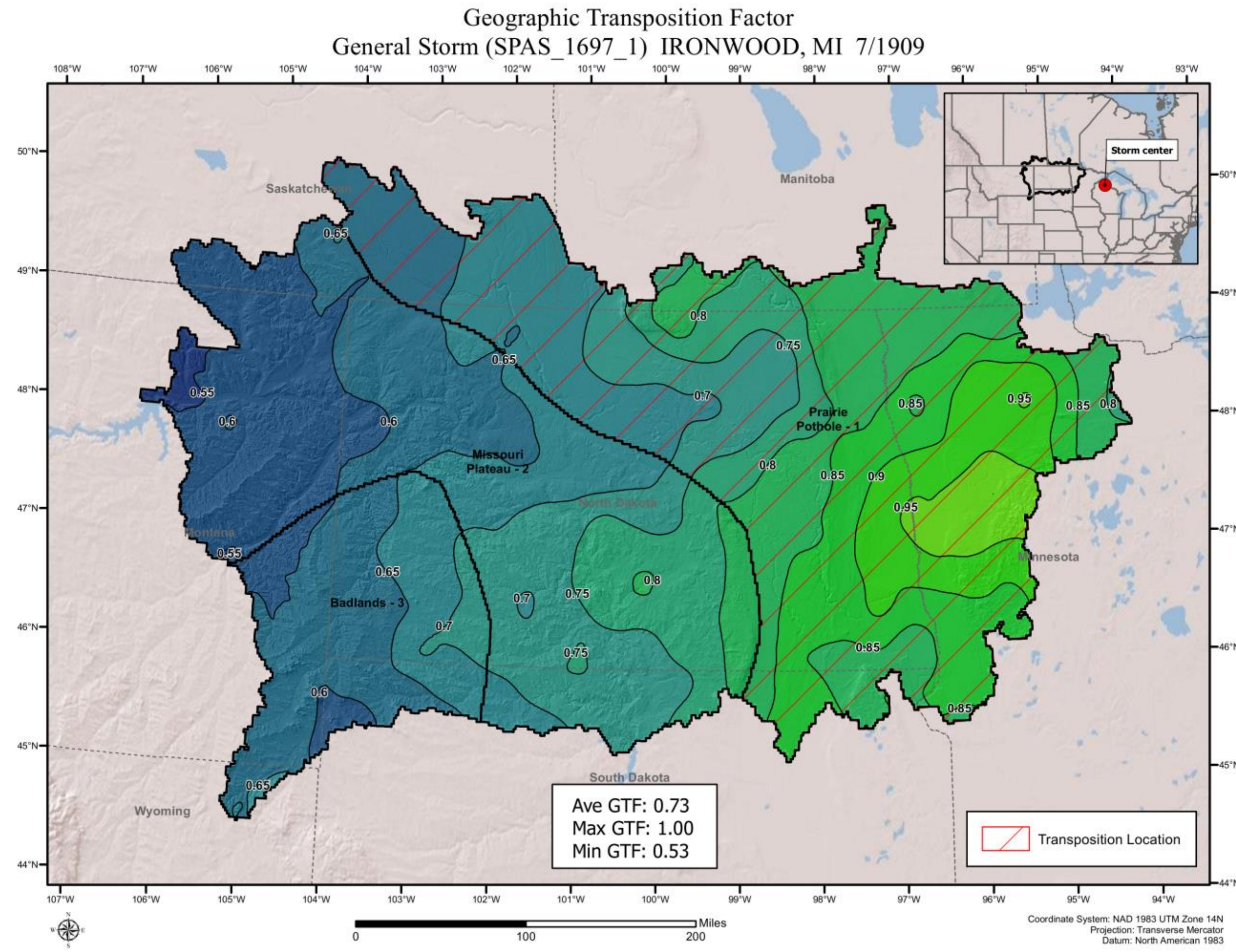


Geographic Transposition Factor
General Storm (SPAS_1583_1) COUNCIL GROVE, KS 7/1951



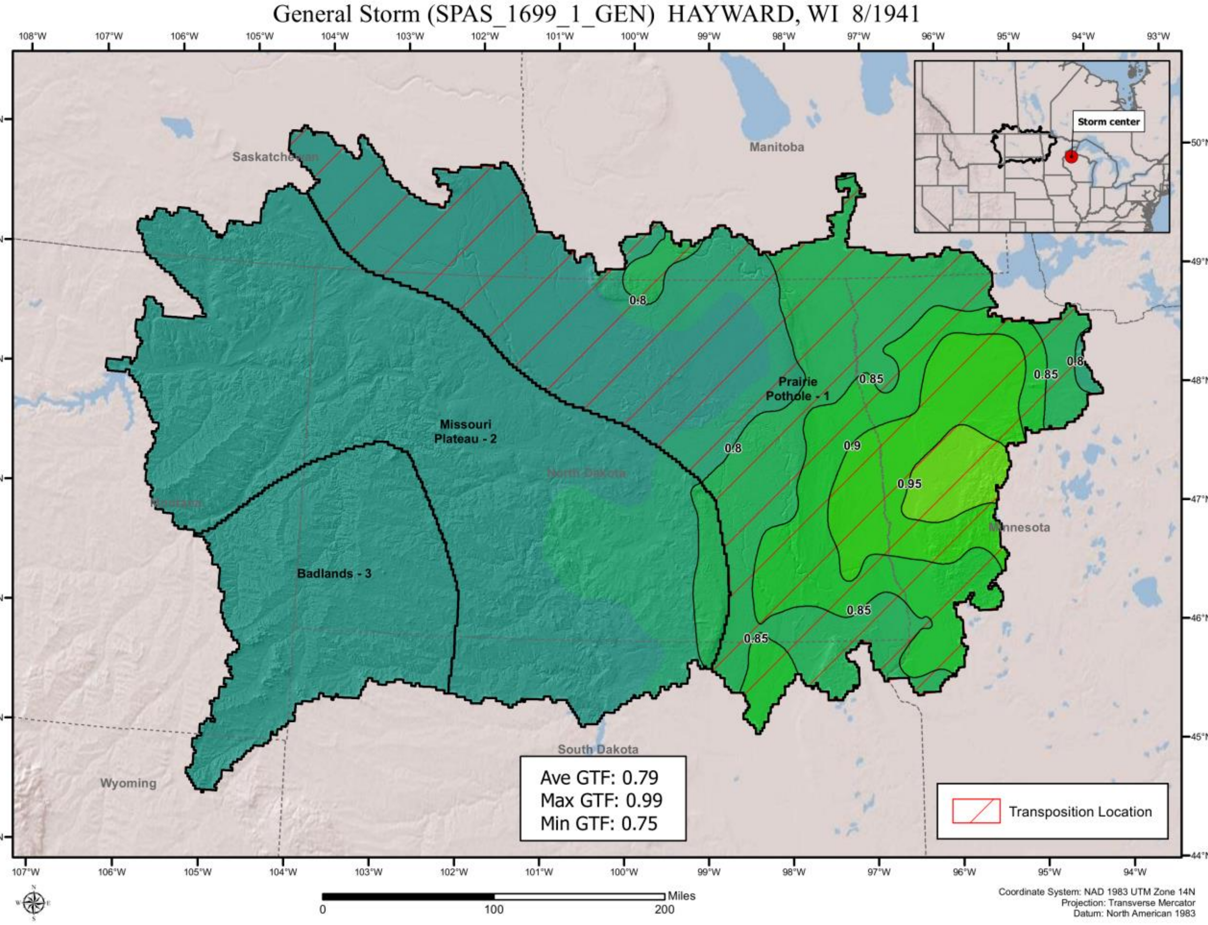
Geographic Transposition Factor
General Storm (SPAS_1630_1) BOLTON, ONT 10/1954



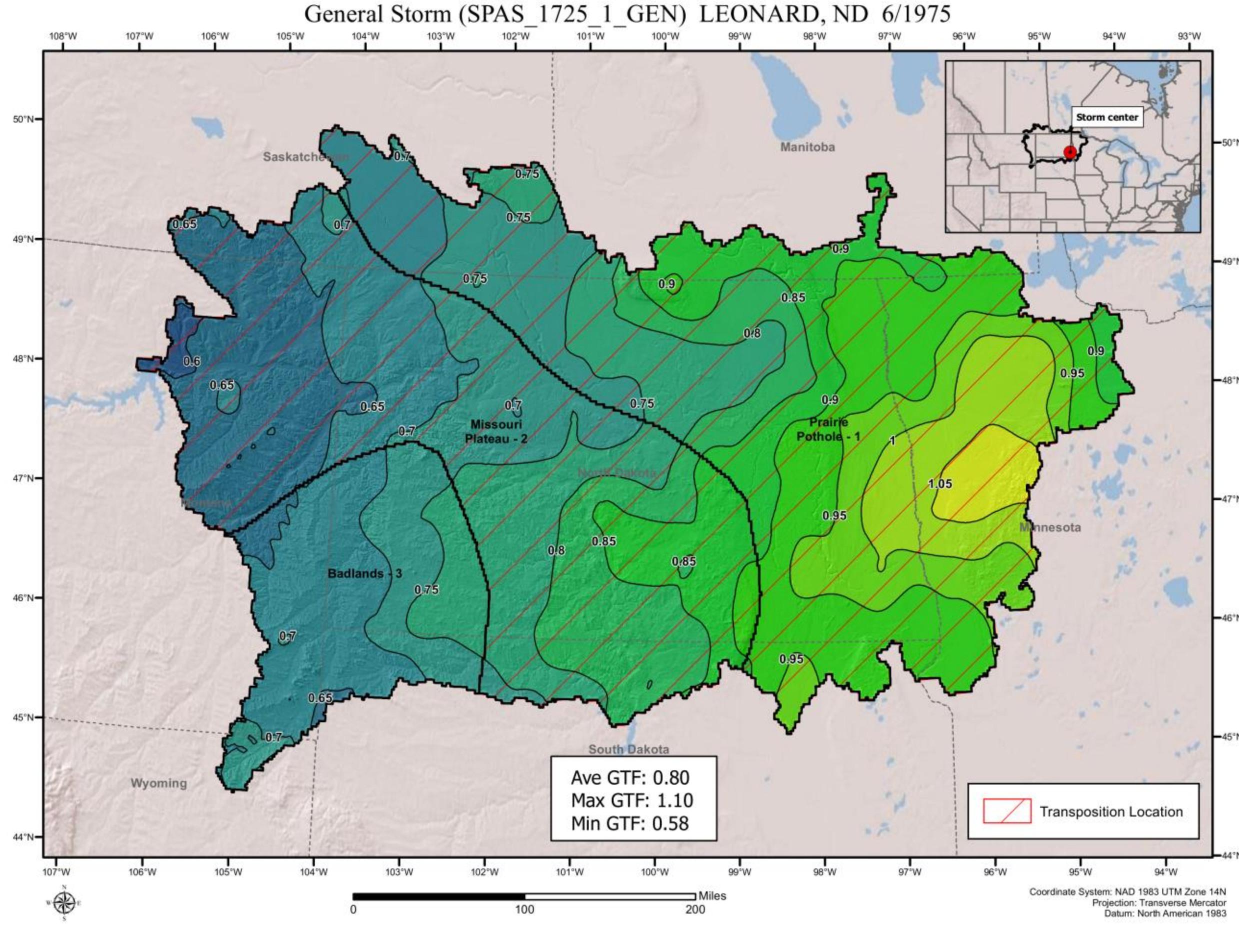


Geographic Transposition Factor
General Storm (SPAS_1697_1) IRONWOOD, MI 7/1909
Storm center
Saskatchewan
Manitoba
Montana
North Dakota
South Dakota
Minnesota
Wyoming
Missouri Plateau - 2
Prairie Pothole - 1
Badlands - 3
Ave GTF: 0.73
Max GTF: 1.00
Min GTF: 0.53
Transposition Location
Miles
0
100
200
Coordinate System: NAD 1983 UTM Zone 14N
Projection: Transverse Mercator
Datum: North American 1983

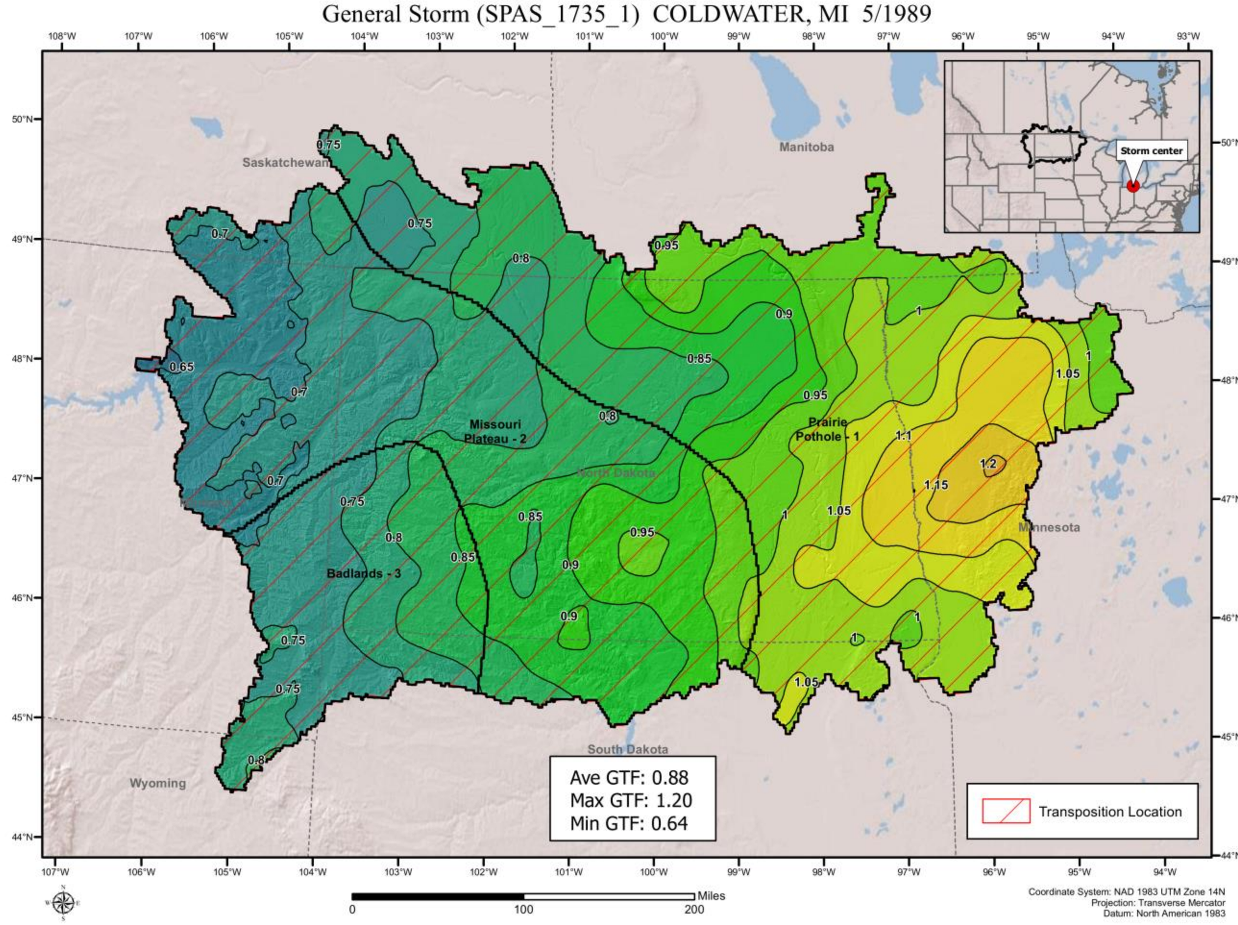
Geographic Transposition Factor
General Storm (SPAS_1699_1_GEN) HAYWARD, WI 8/1941



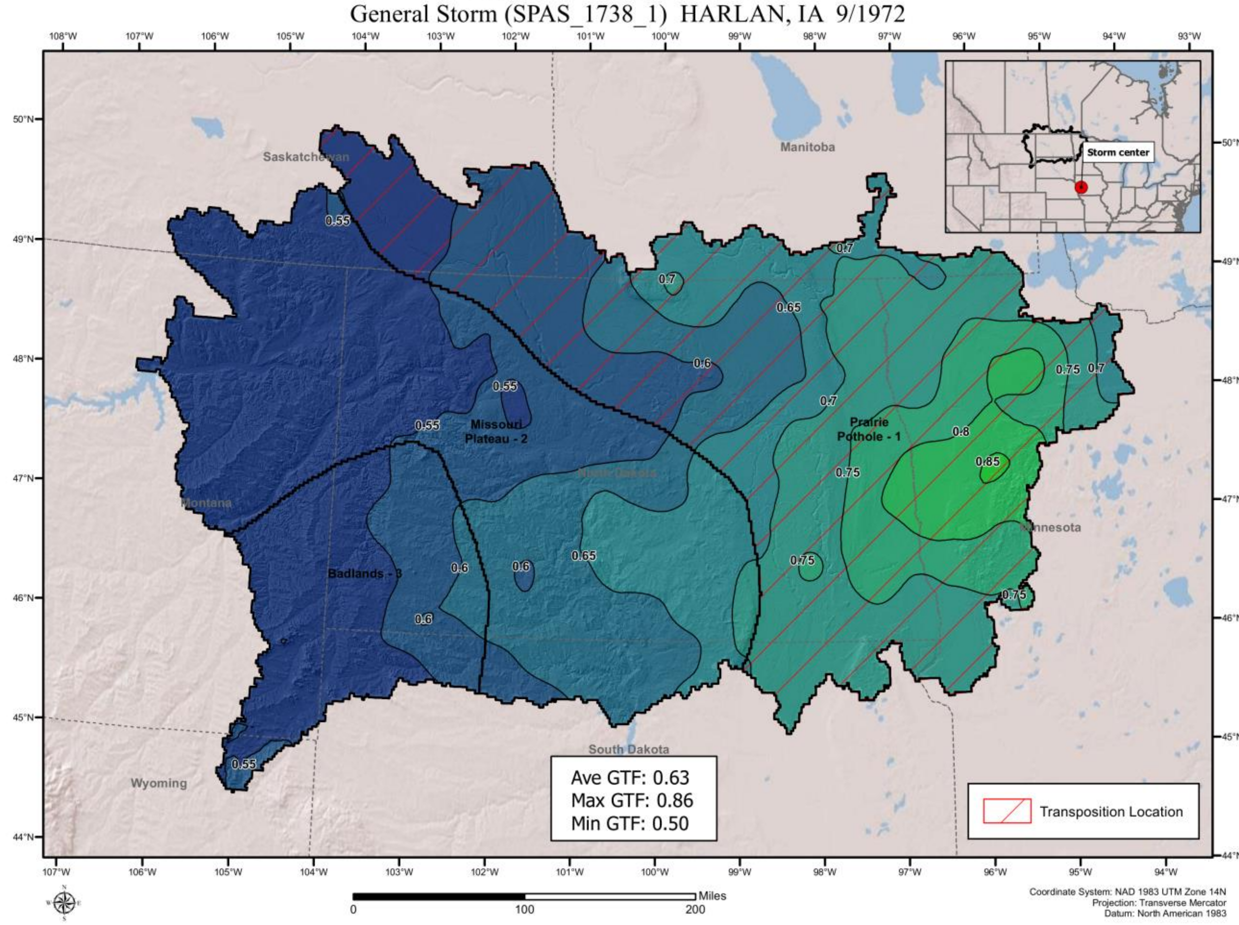
Geographic Transposition Factor
General Storm (SPAS_1725_1_GEN) LEONARD, ND 6/1975



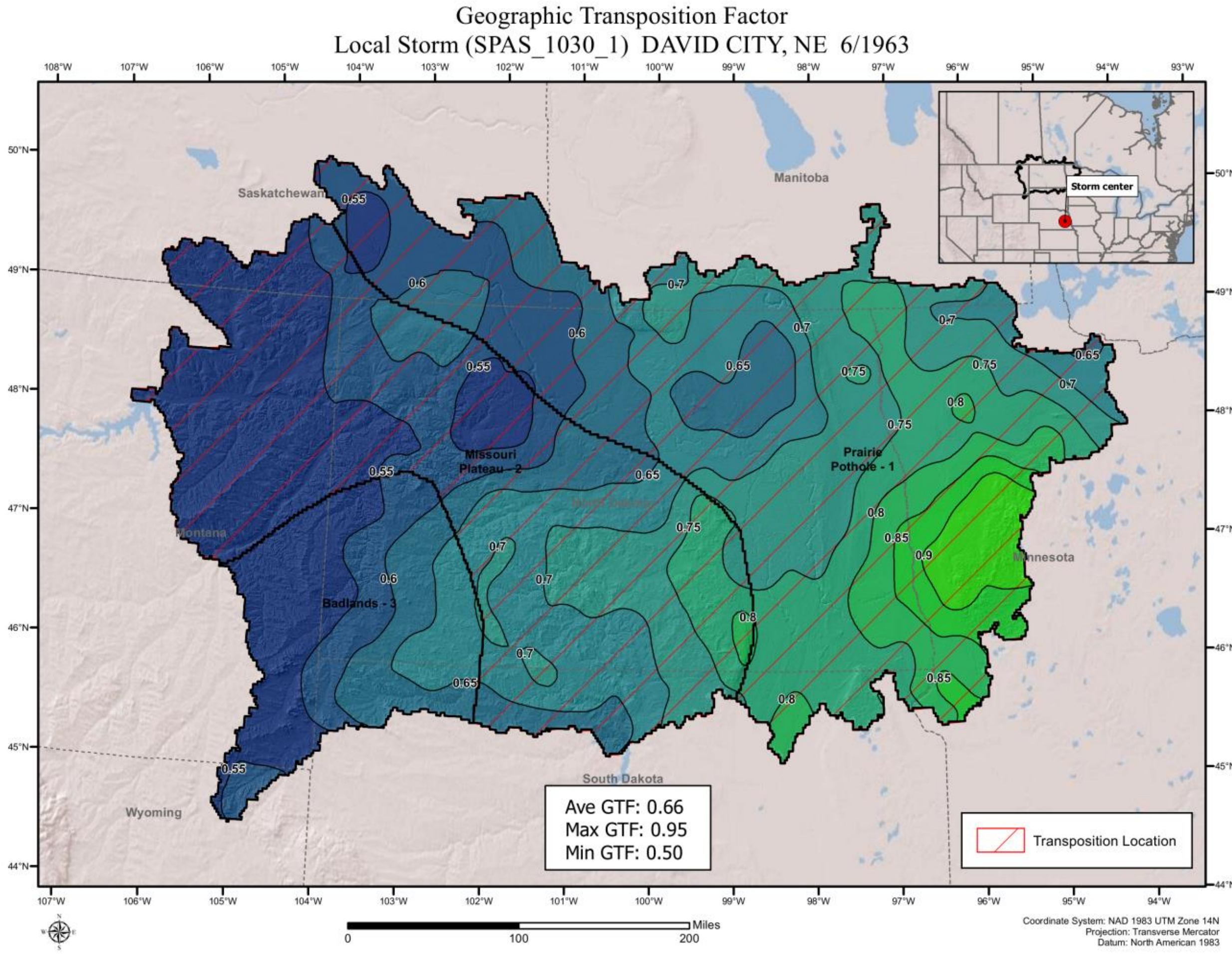
Geographic Transposition Factor
General Storm (SPAS_1735_1) COLDWATER, MI 5/1989



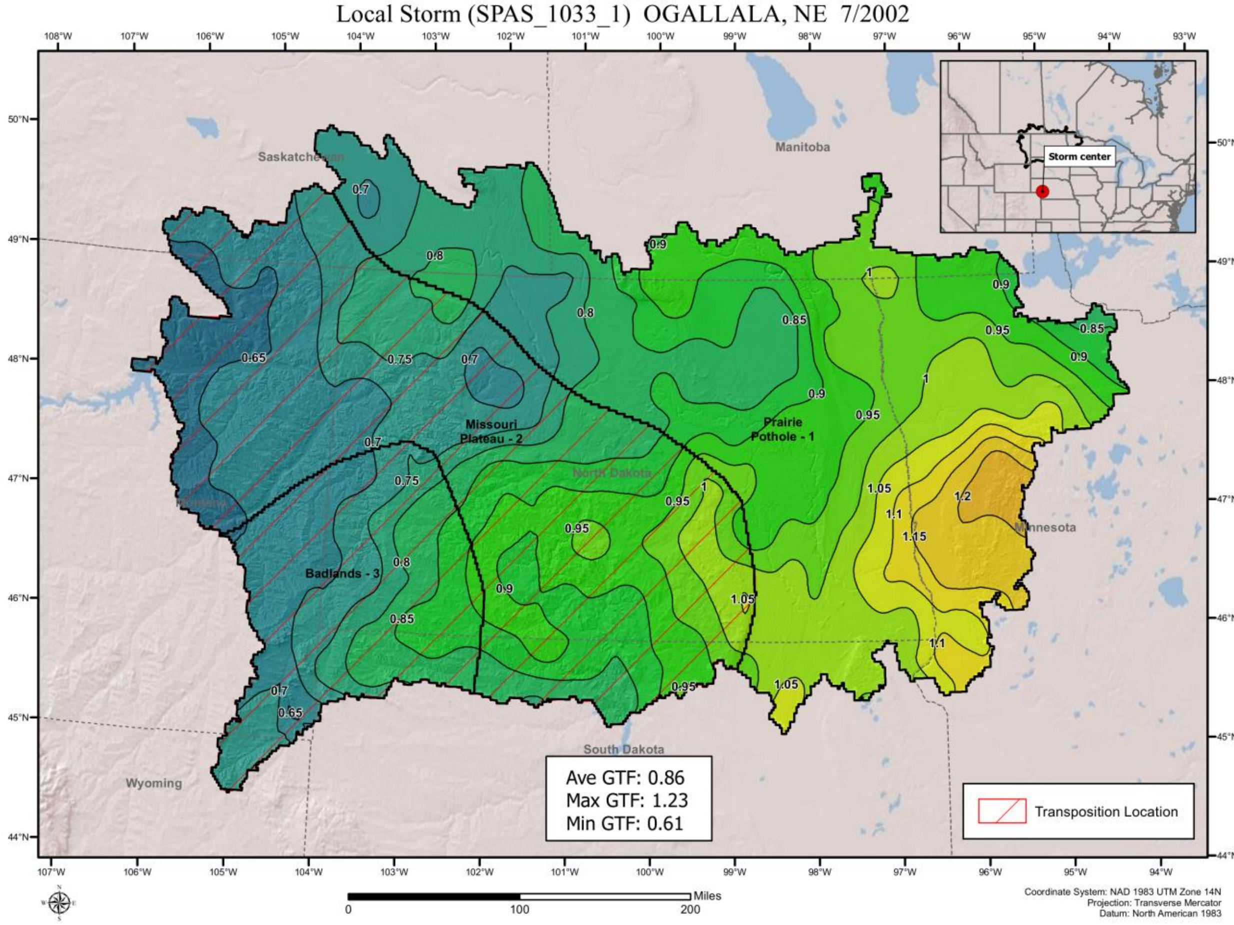
Geographic Transposition Factor
General Storm (SPAS_1738_1) HARLAN, IA 9/1972



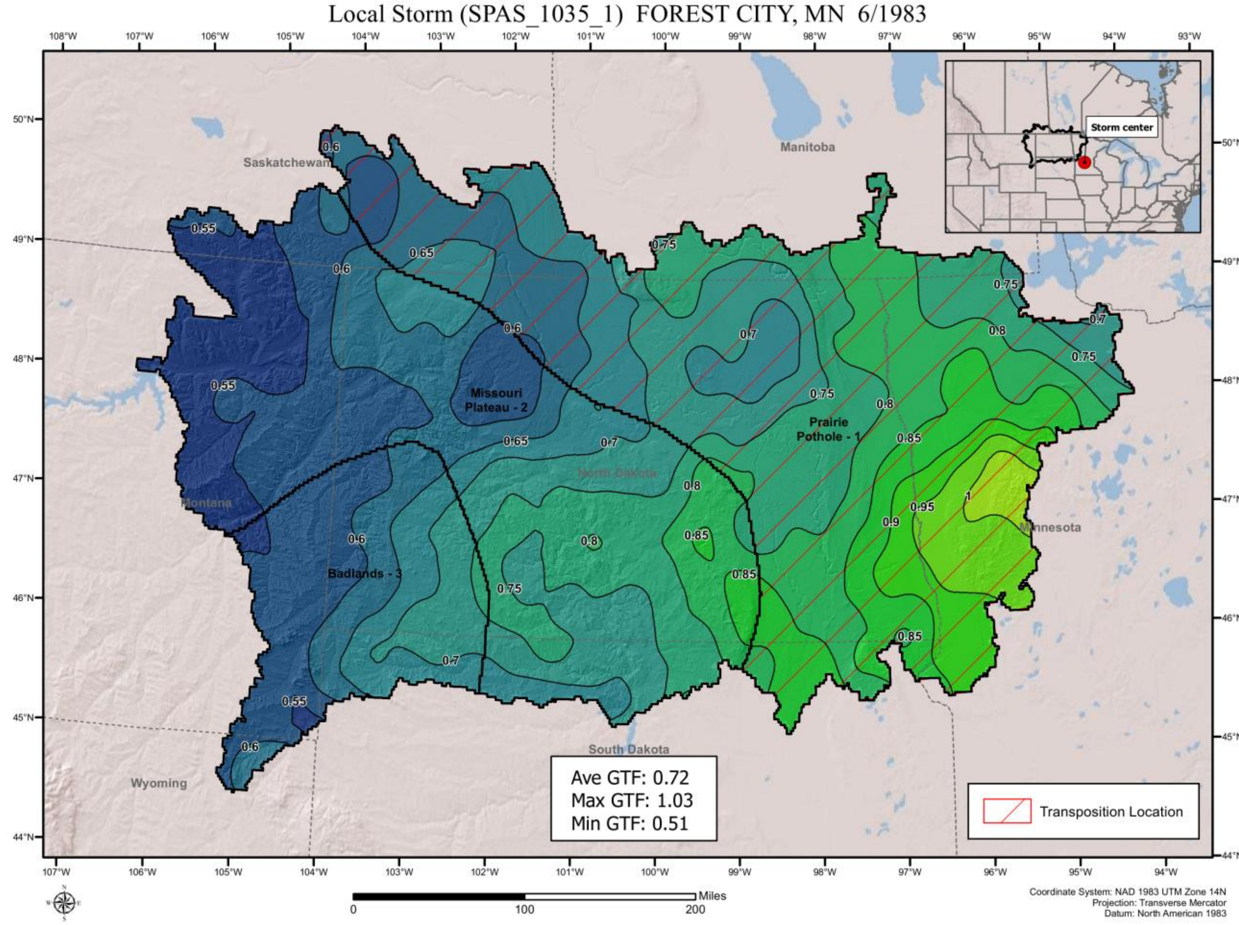
Local Storms



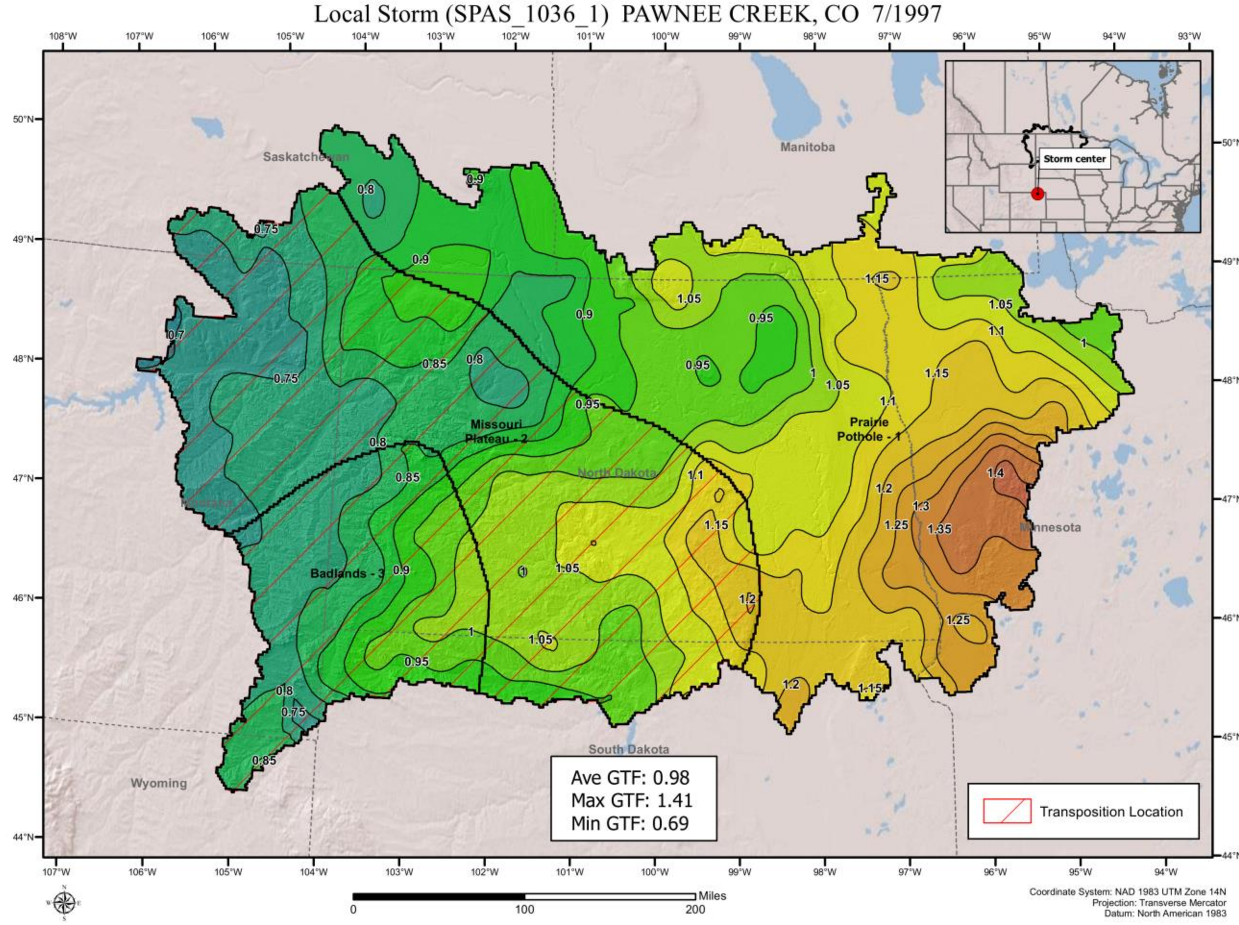
Geographic Transposition Factor
Local Storm (SPAS_1033_1) OGALLALA, NE 7/2002



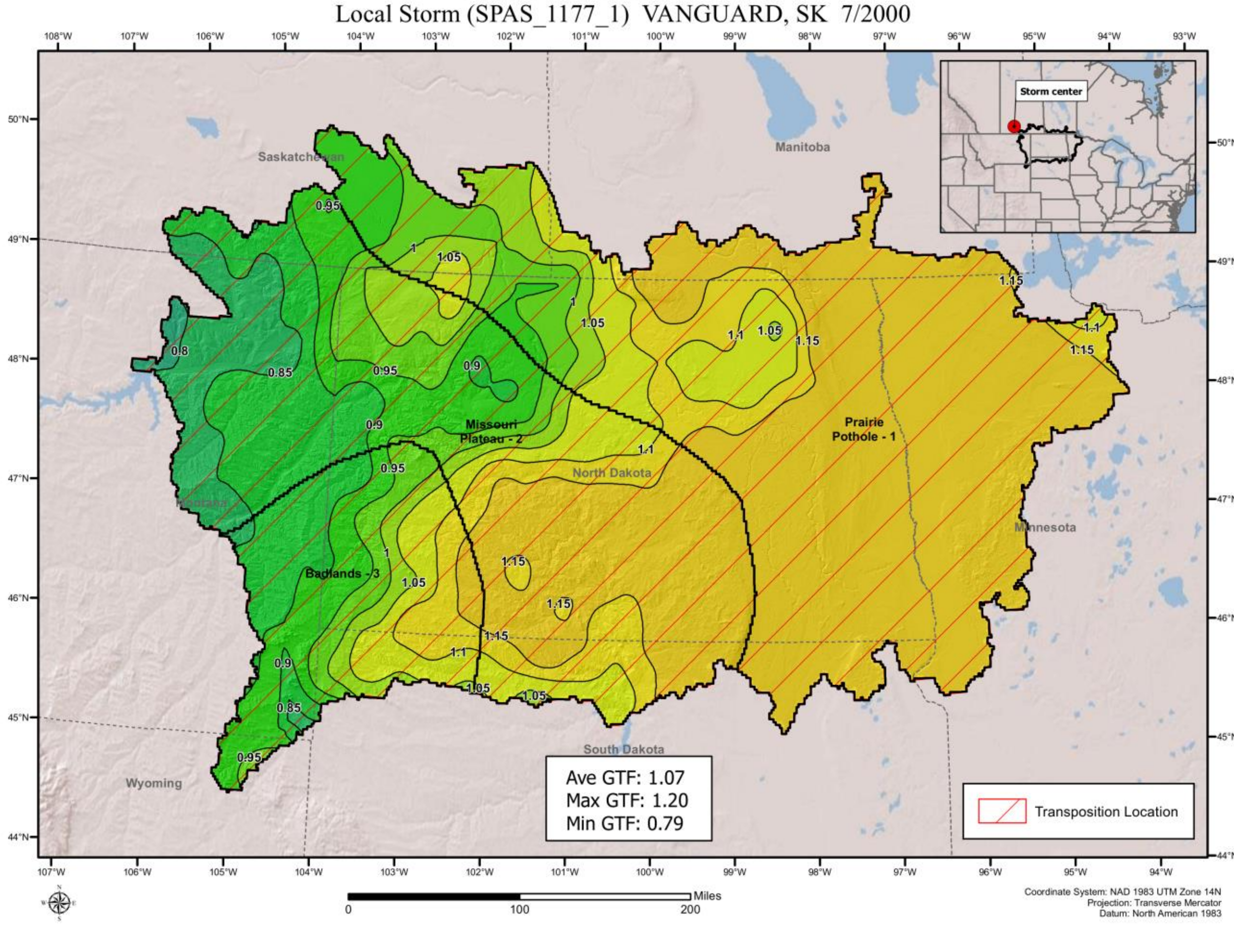
Geographic Transposition Factor
Local Storm (SPAS_1035_1) FOREST CITY, MN 6/1983



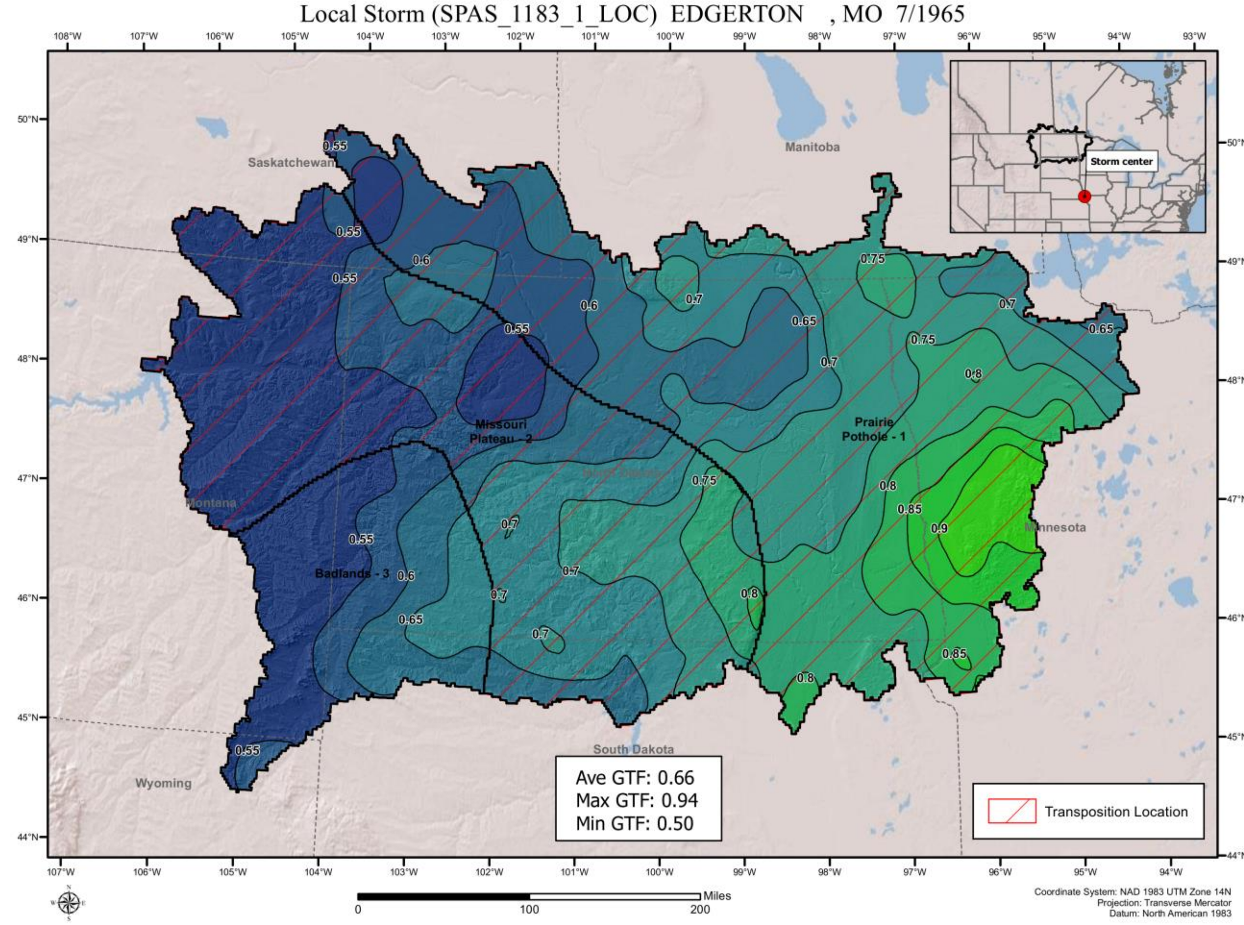
Geographic Transposition Factor
Local Storm (SPAS_1036_1) PAWNEE CREEK, CO 7/1997



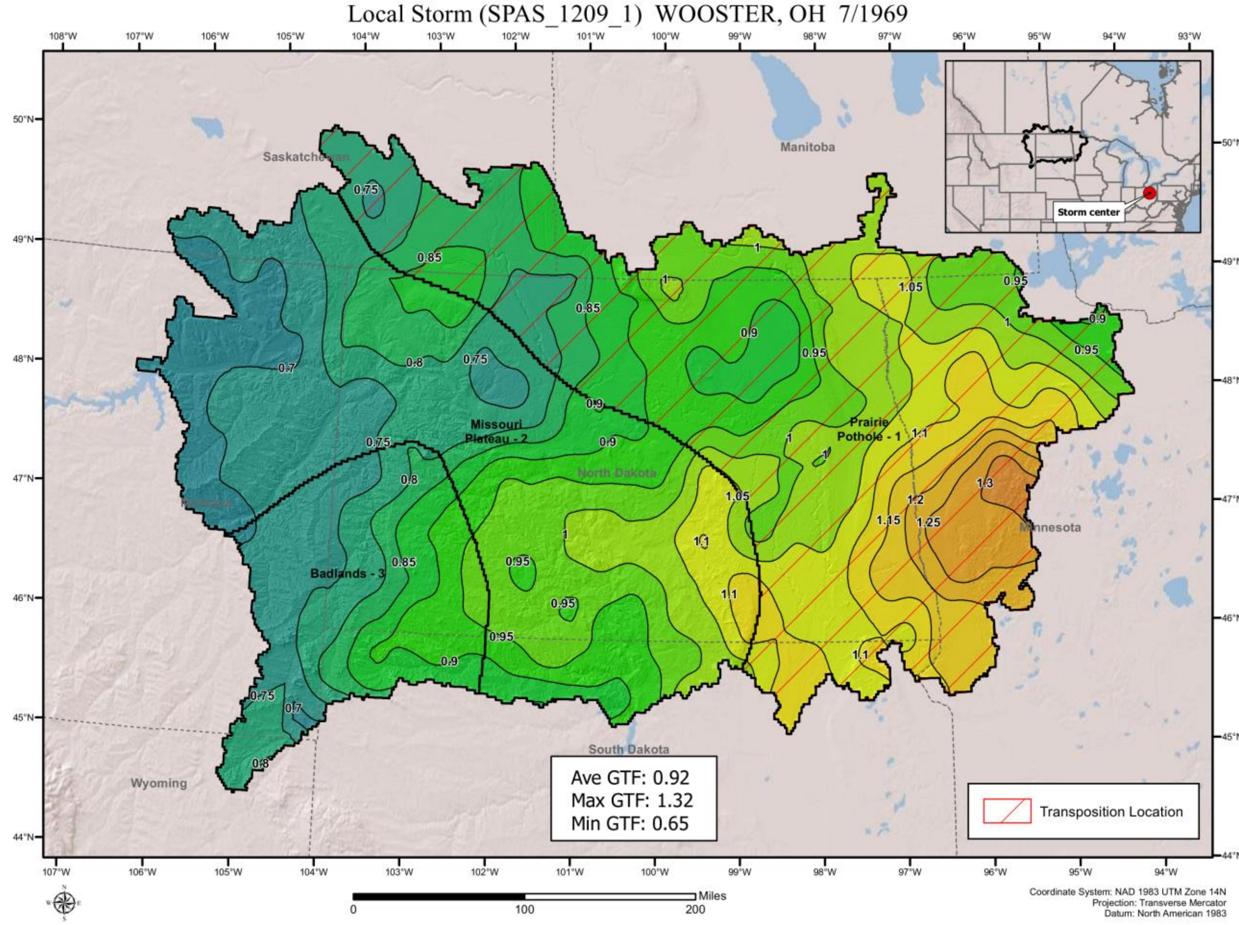
Geographic Transposition Factor
Local Storm (SPAS_1177_1) VANGUARD, SK 7/2000



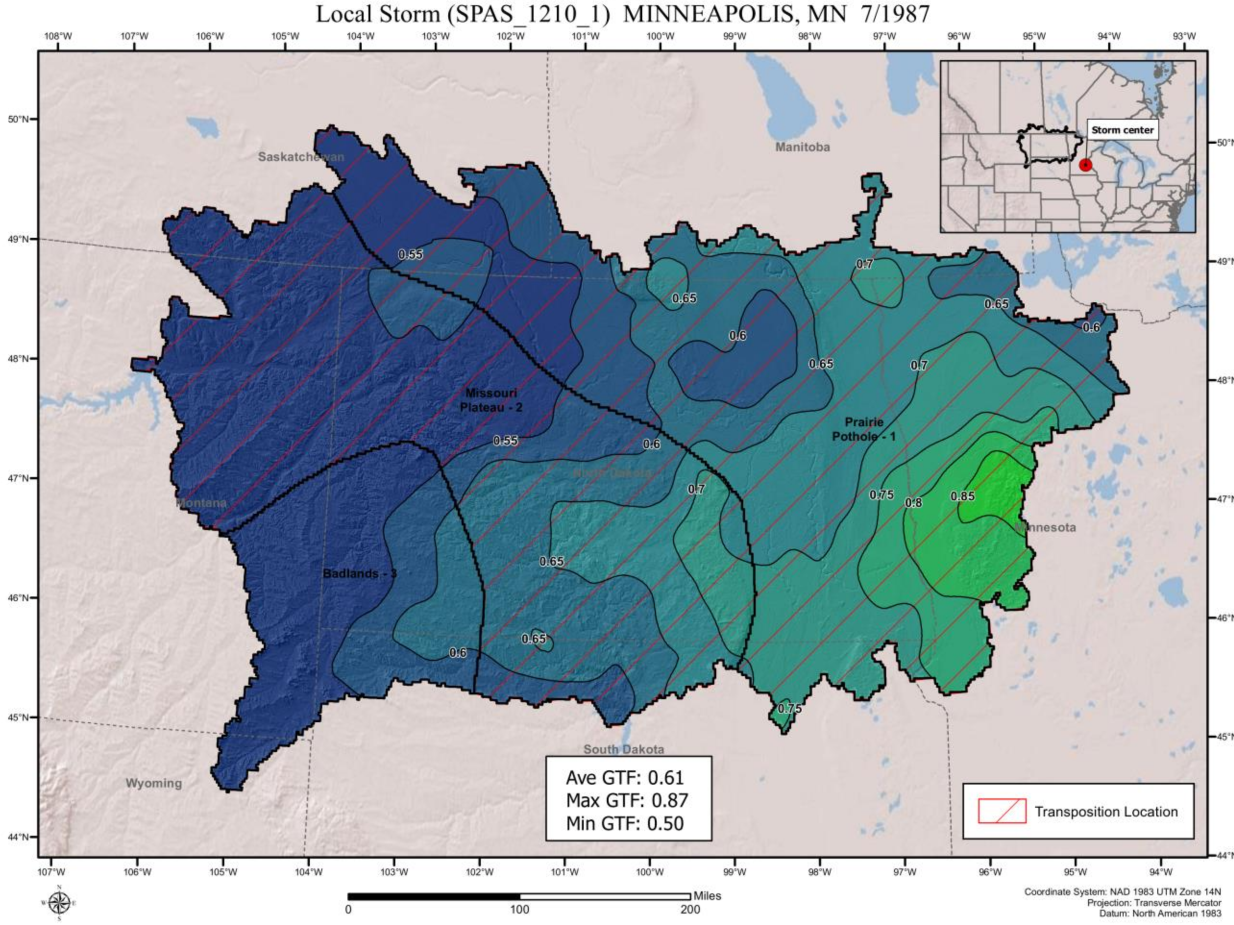
Geographic Transposition Factor
Local Storm (SPAS_1183_1_LOC) EDGERTON , MO 7/1965



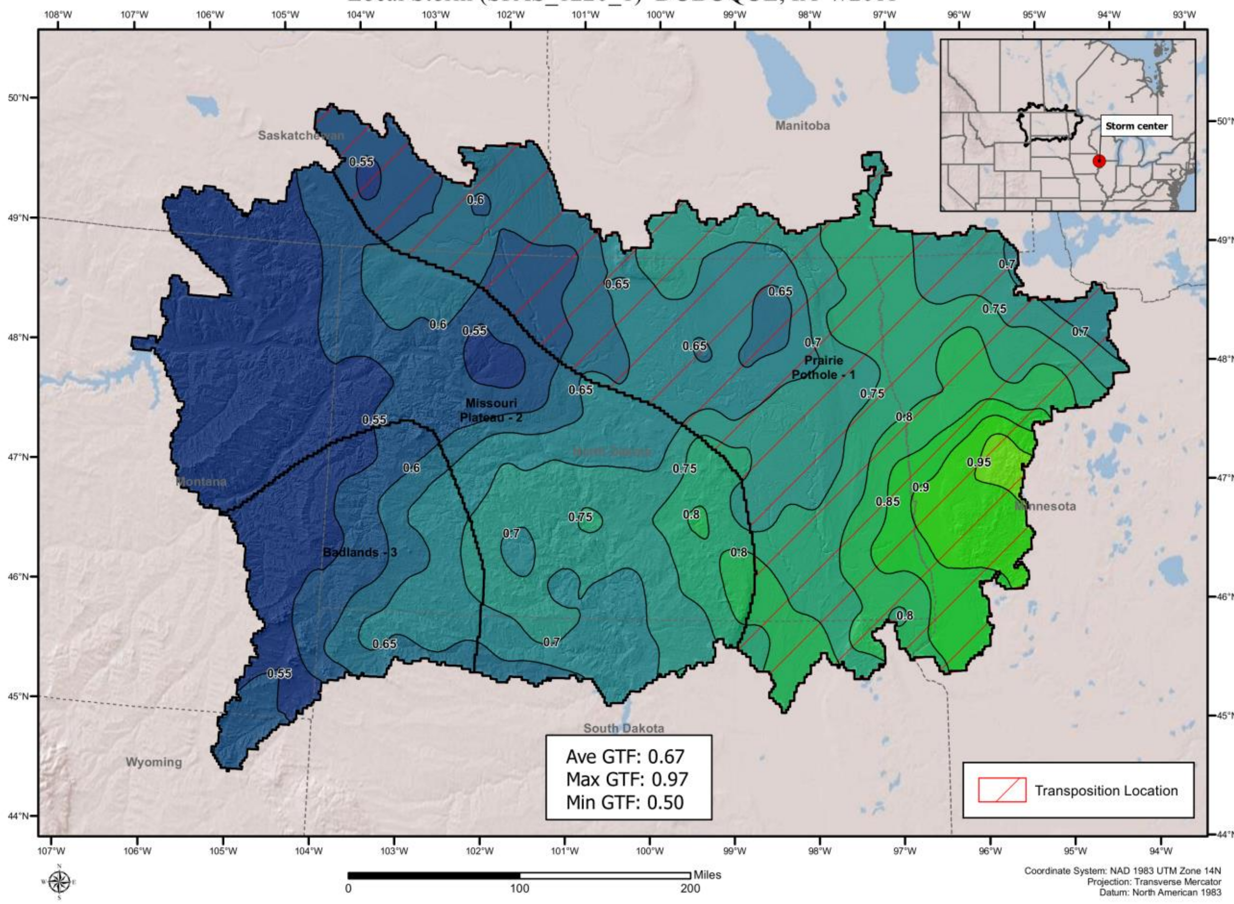
Geographic Transposition Factor
Local Storm (SPAS_1209_1) WOOSTER, OH 7/1969



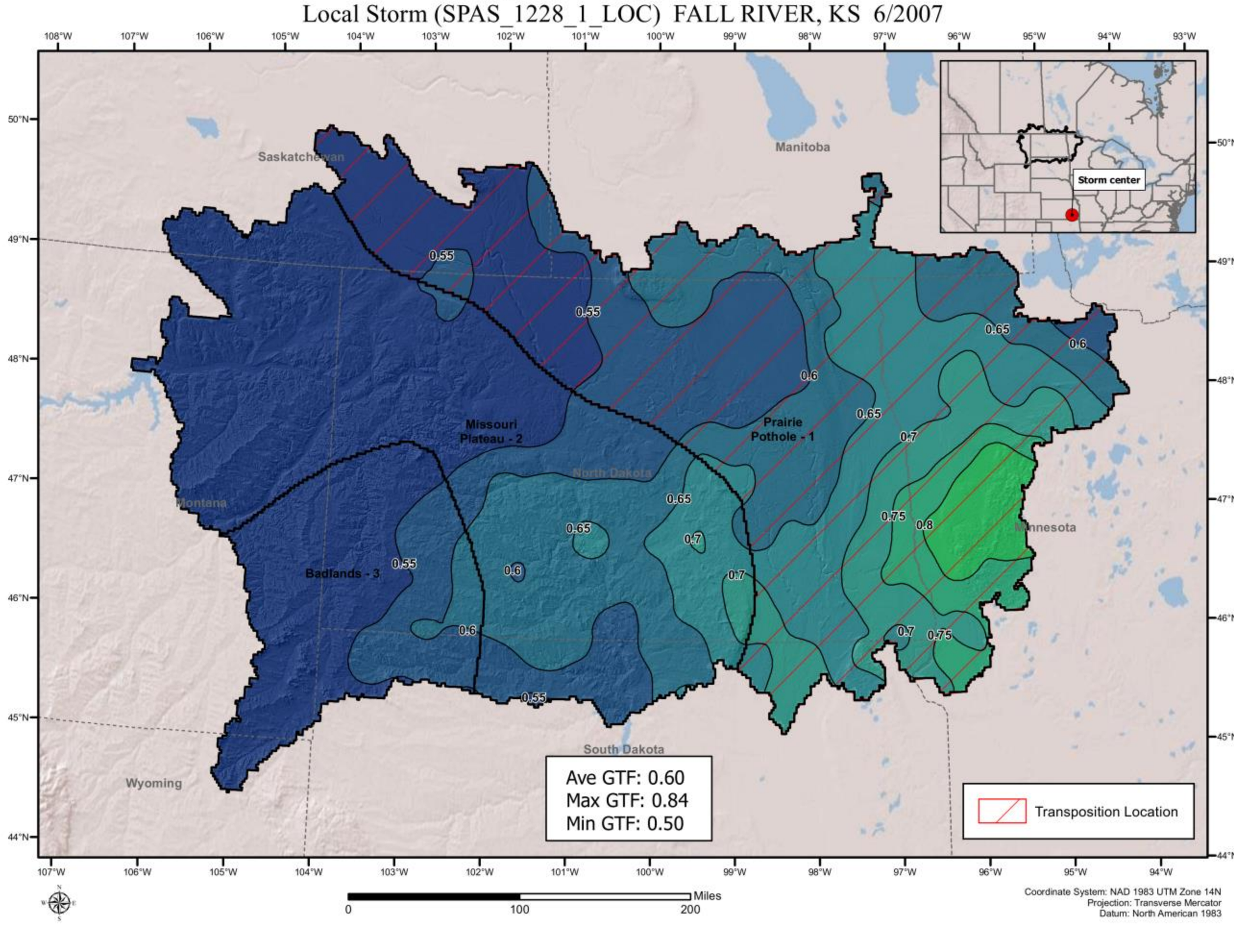
Geographic Transposition Factor
Local Storm (SPAS_1210_1) MINNEAPOLIS, MN 7/1987



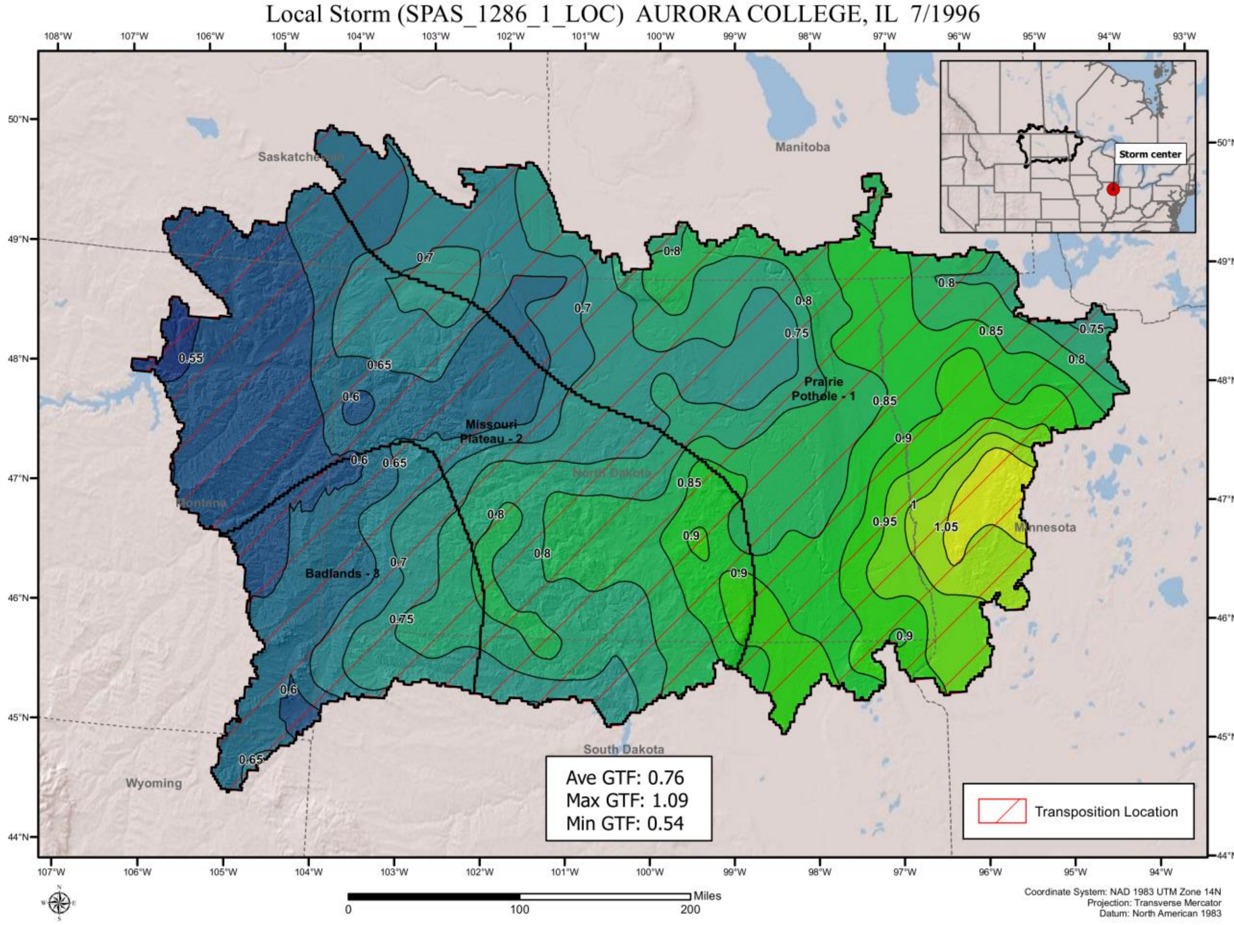
Geographic Transposition Factor
Local Storm (SPAS_1220_1) DUBUQUE, IA 7/2011



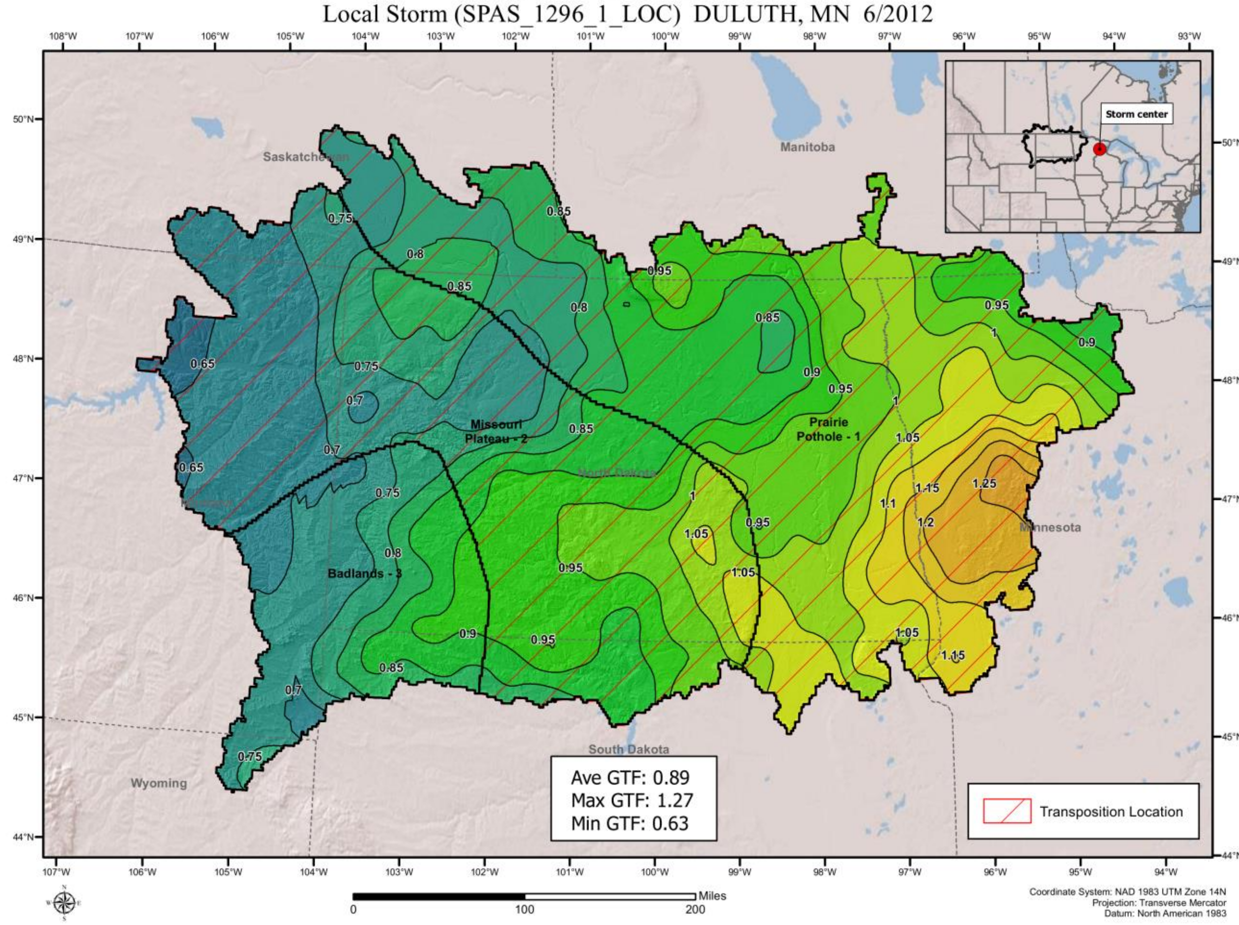
Geographic Transposition Factor
Local Storm (SPAS_1228_1_LOC) FALL RIVER, KS 6/2007



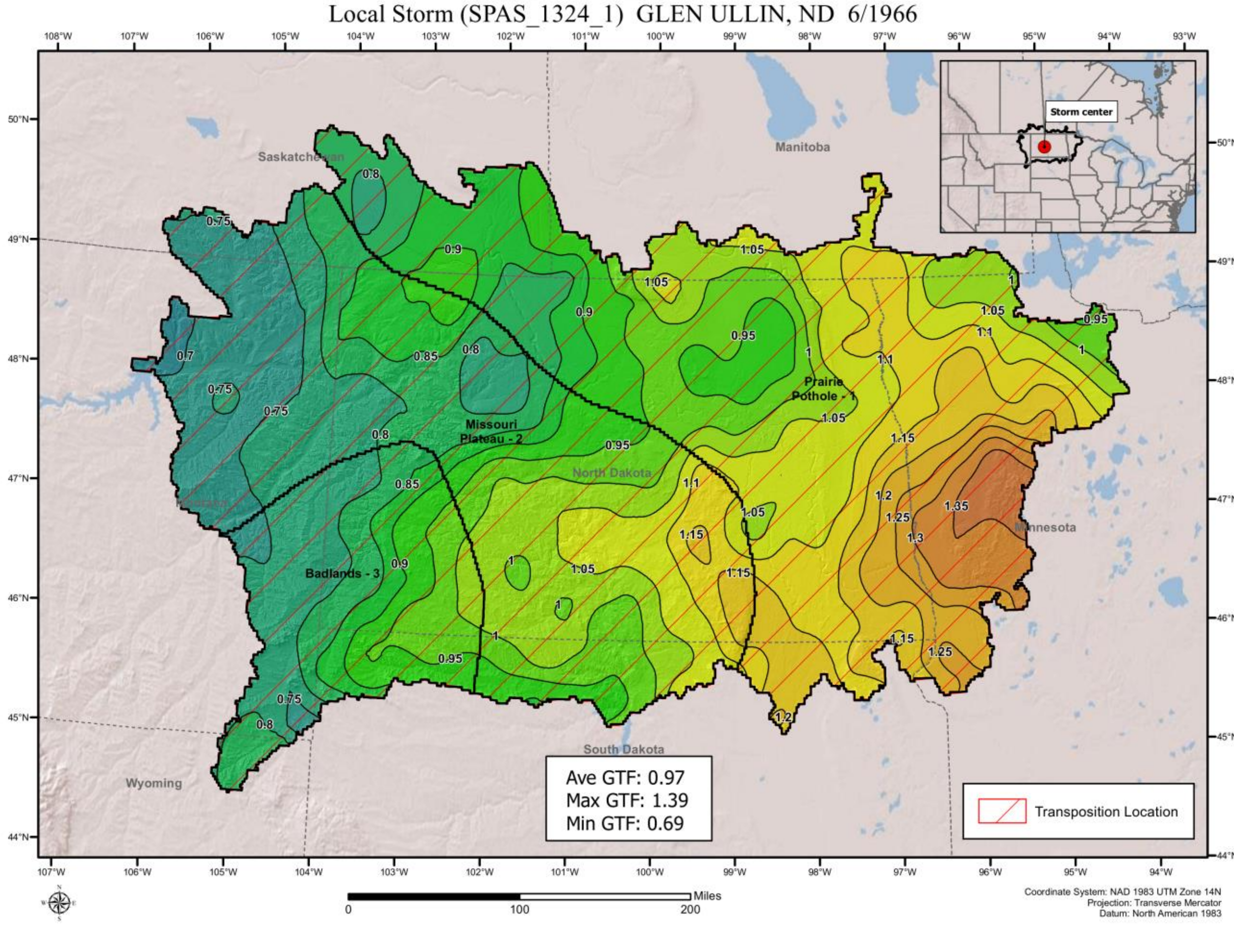
Geographic Transposition Factor
Local Storm (SPAS_1286_1_LOC) AURORA COLLEGE, IL 7/1996



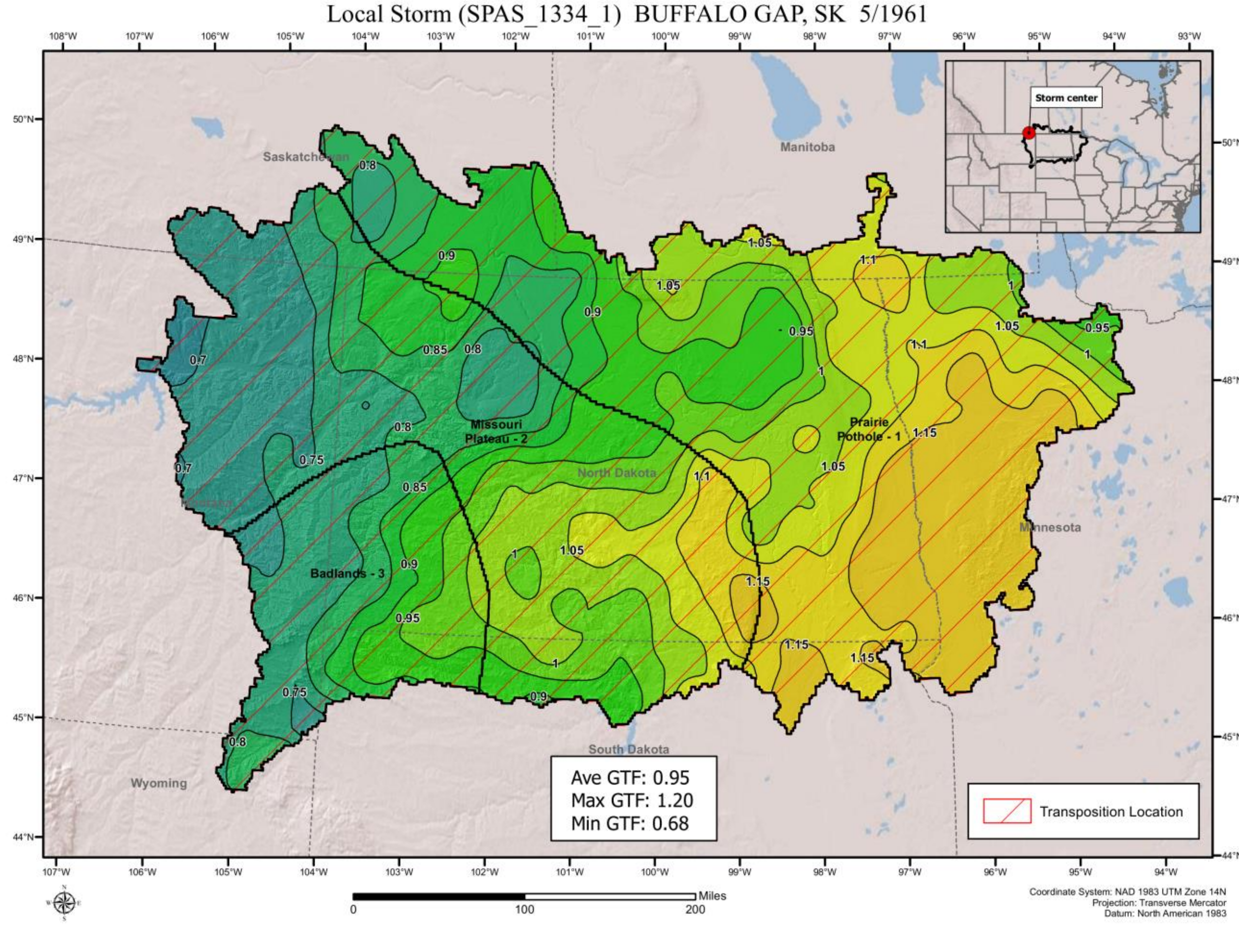
Geographic Transposition Factor
Local Storm (SPAS_1296_1_LOC) DULUTH, MN 6/2012



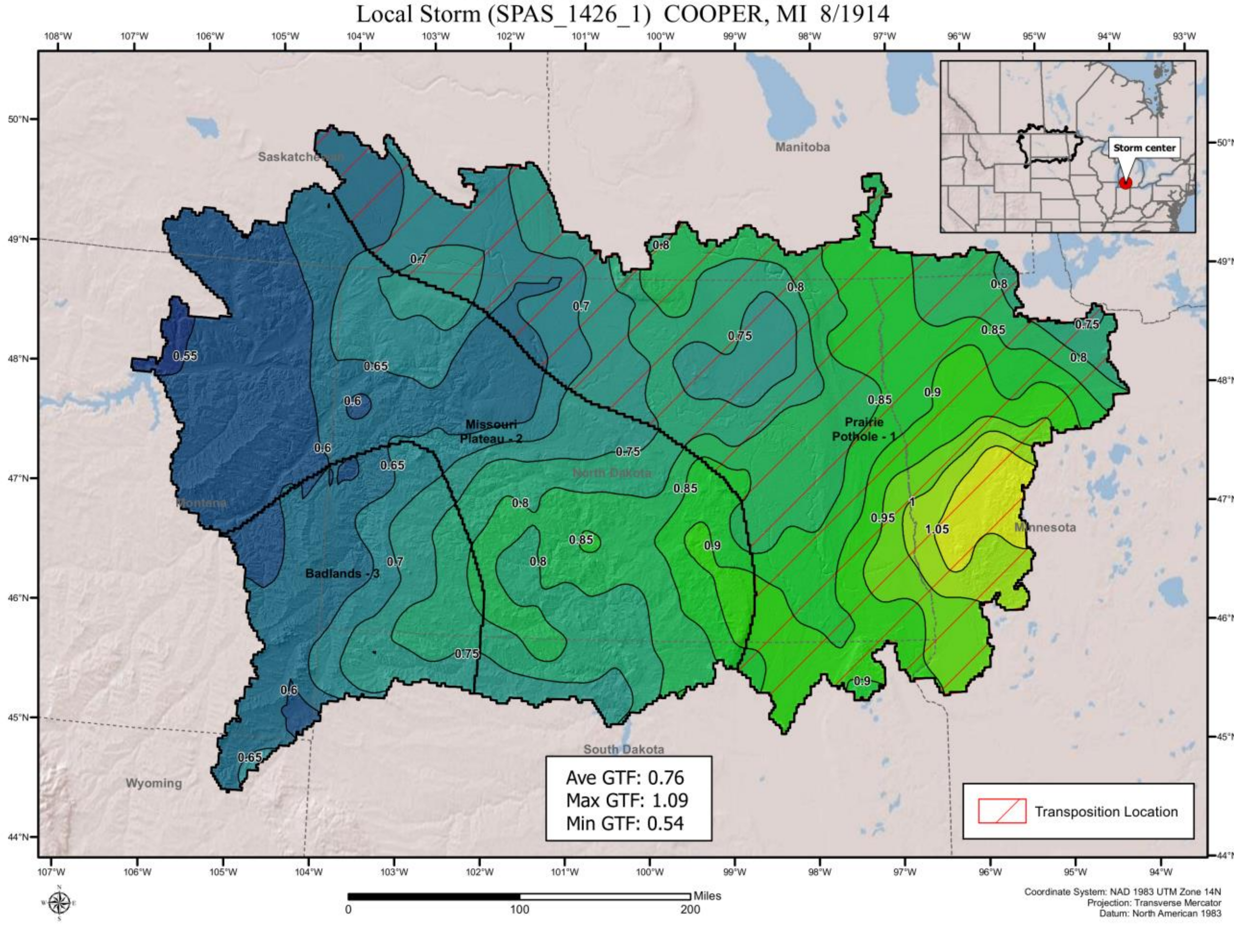
Geographic Transposition Factor
Local Storm (SPAS_1324_1) GLEN ULLIN, ND 6/1966



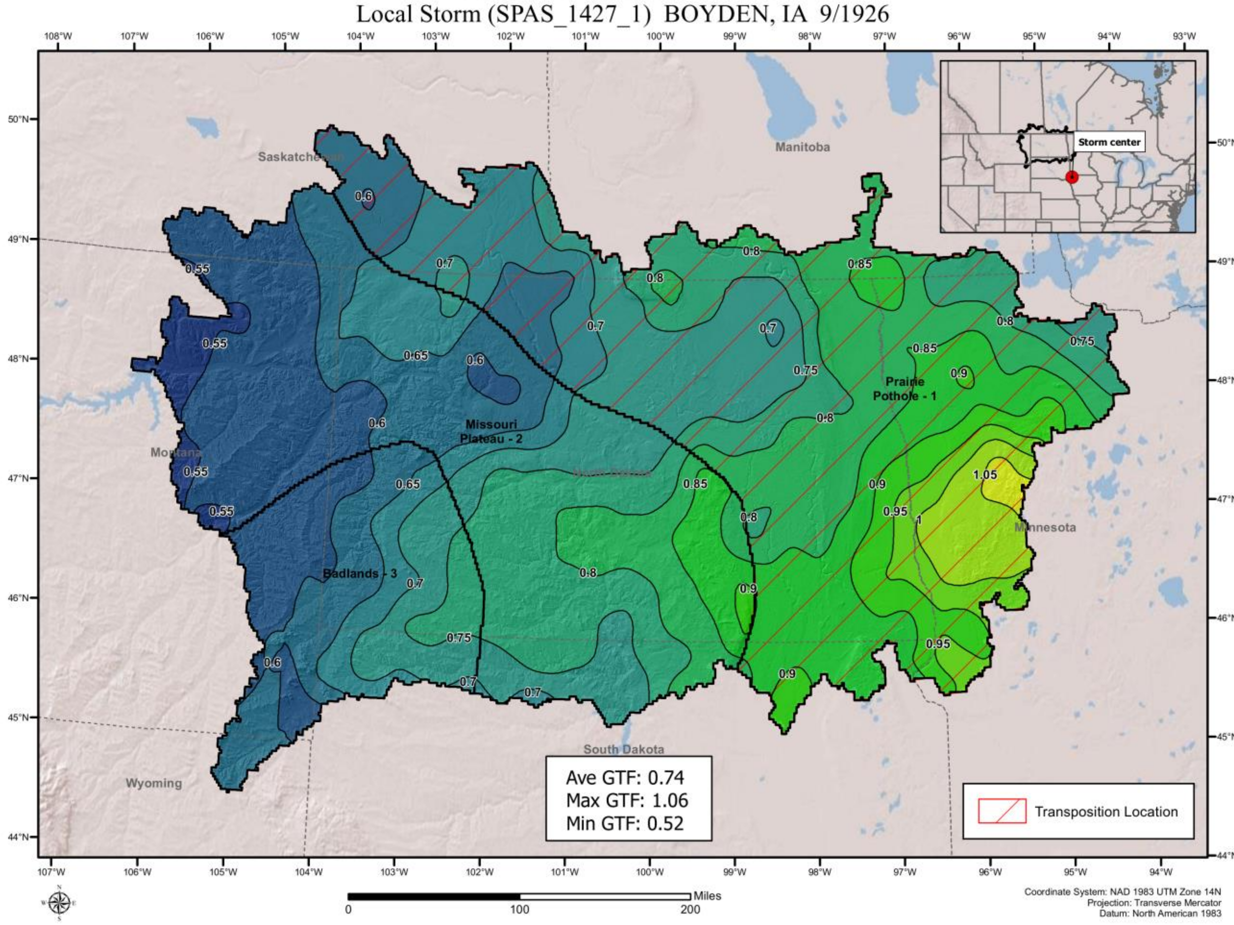
Geographic Transposition Factor
Local Storm (SPAS_1334_1) BUFFALO GAP, SK 5/1961



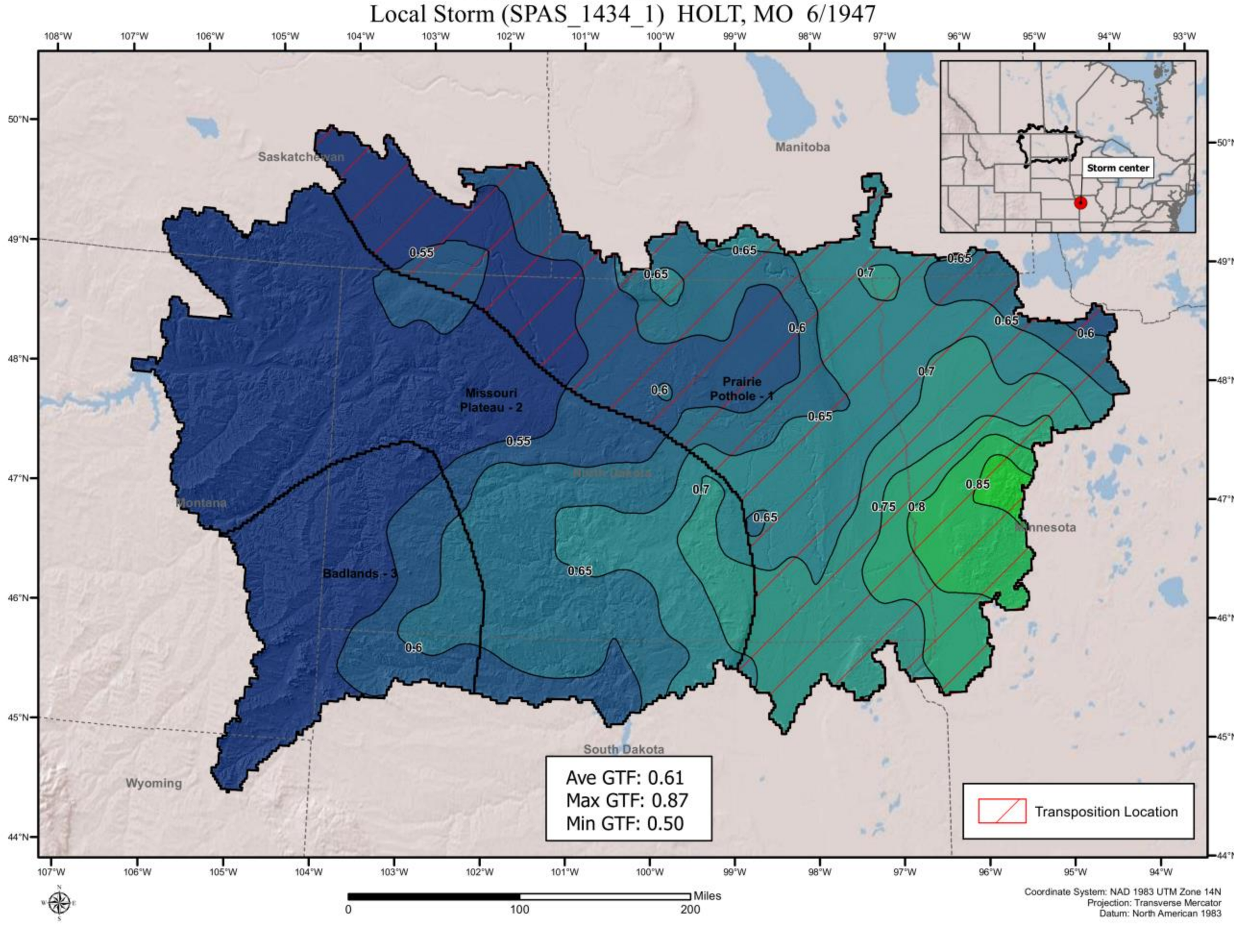
Geographic Transposition Factor
Local Storm (SPAS_1426_1) COOPER, MI 8/1914



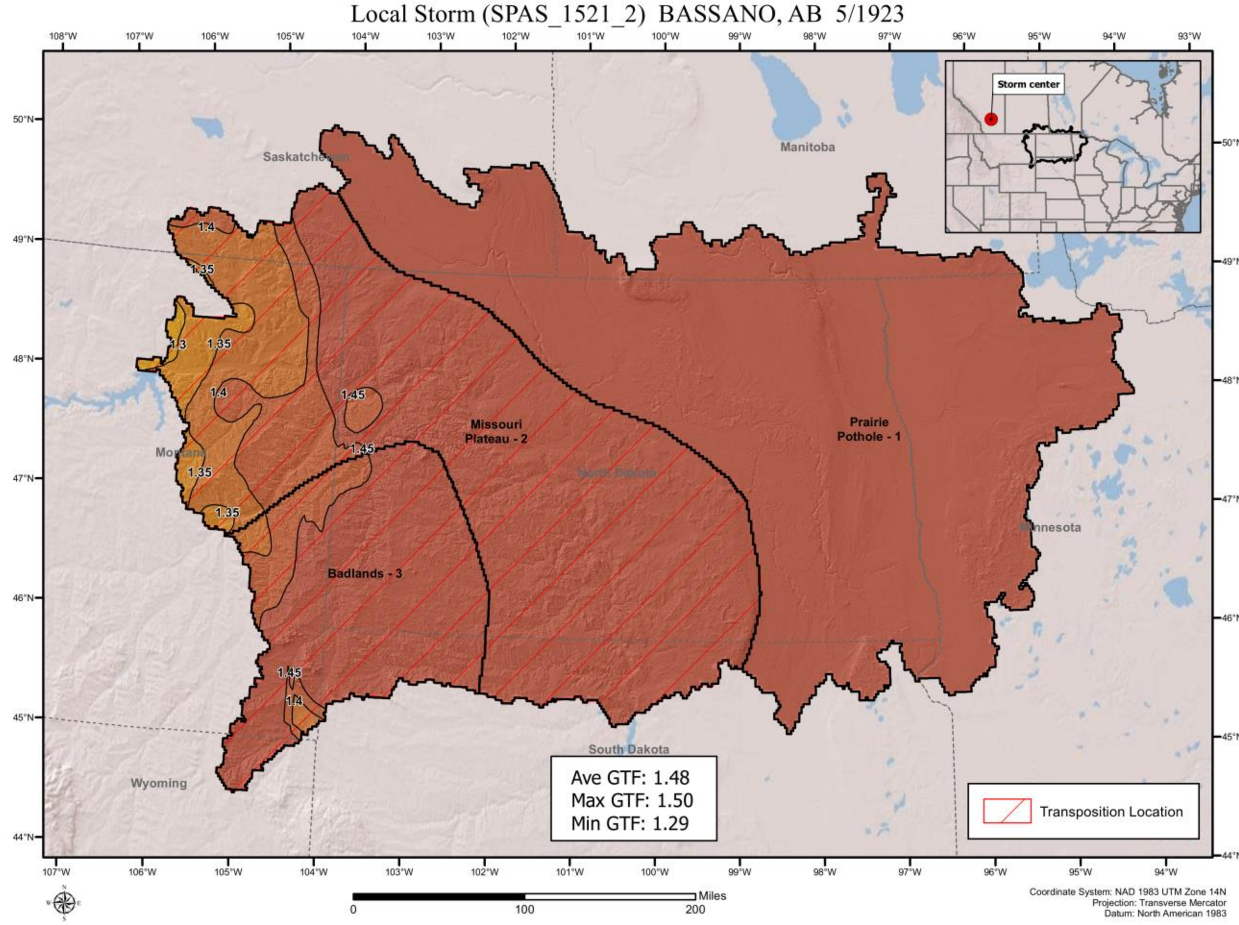
Geographic Transposition Factor
Local Storm (SPAS_1427_1) BOYDEN, IA 9/1926



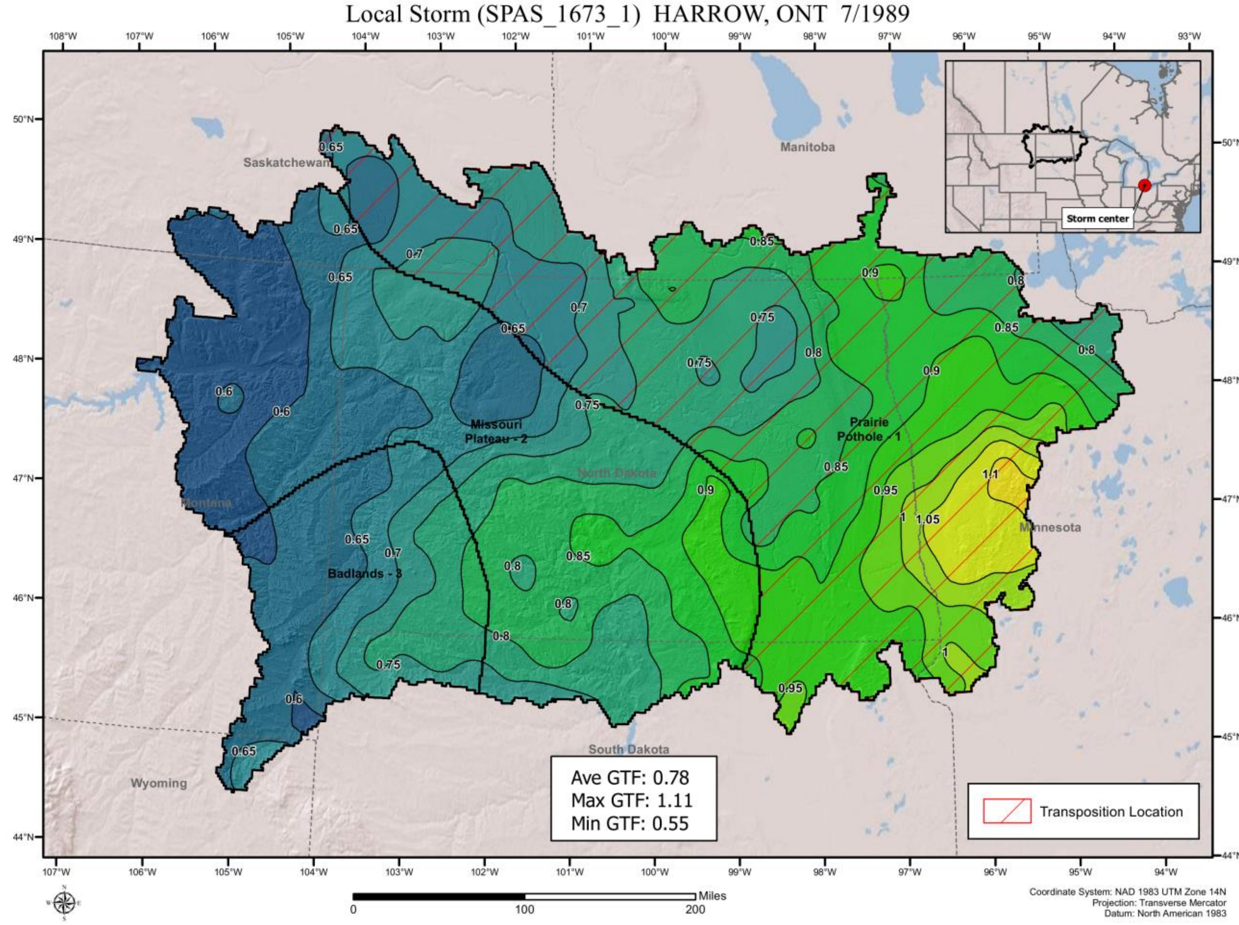
Geographic Transposition Factor
Local Storm (SPAS_1434_1) HOLT, MO 6/1947



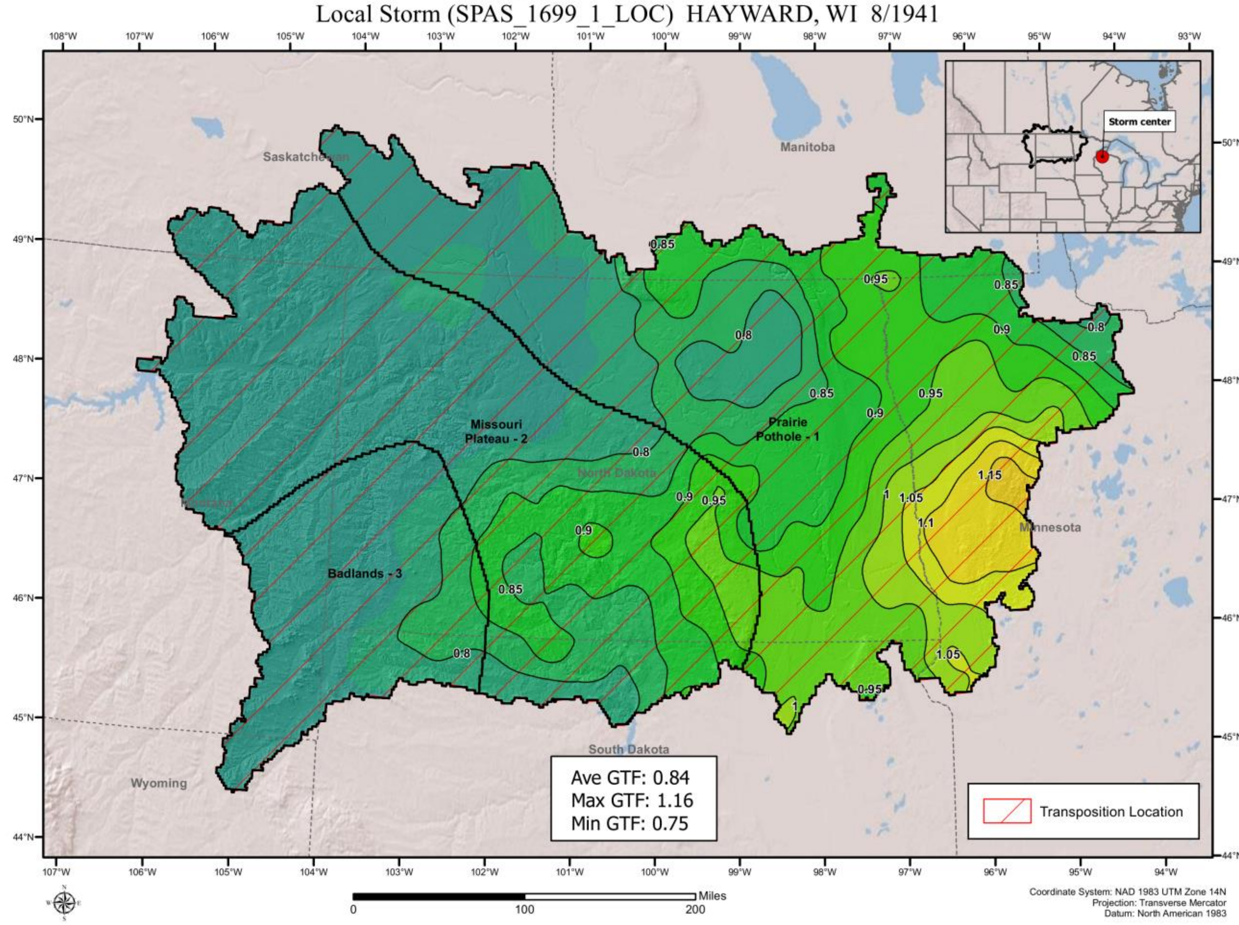
Geographic Transposition Factor
Local Storm (SPAS_1521_2) BASSANO, AB 5/1923



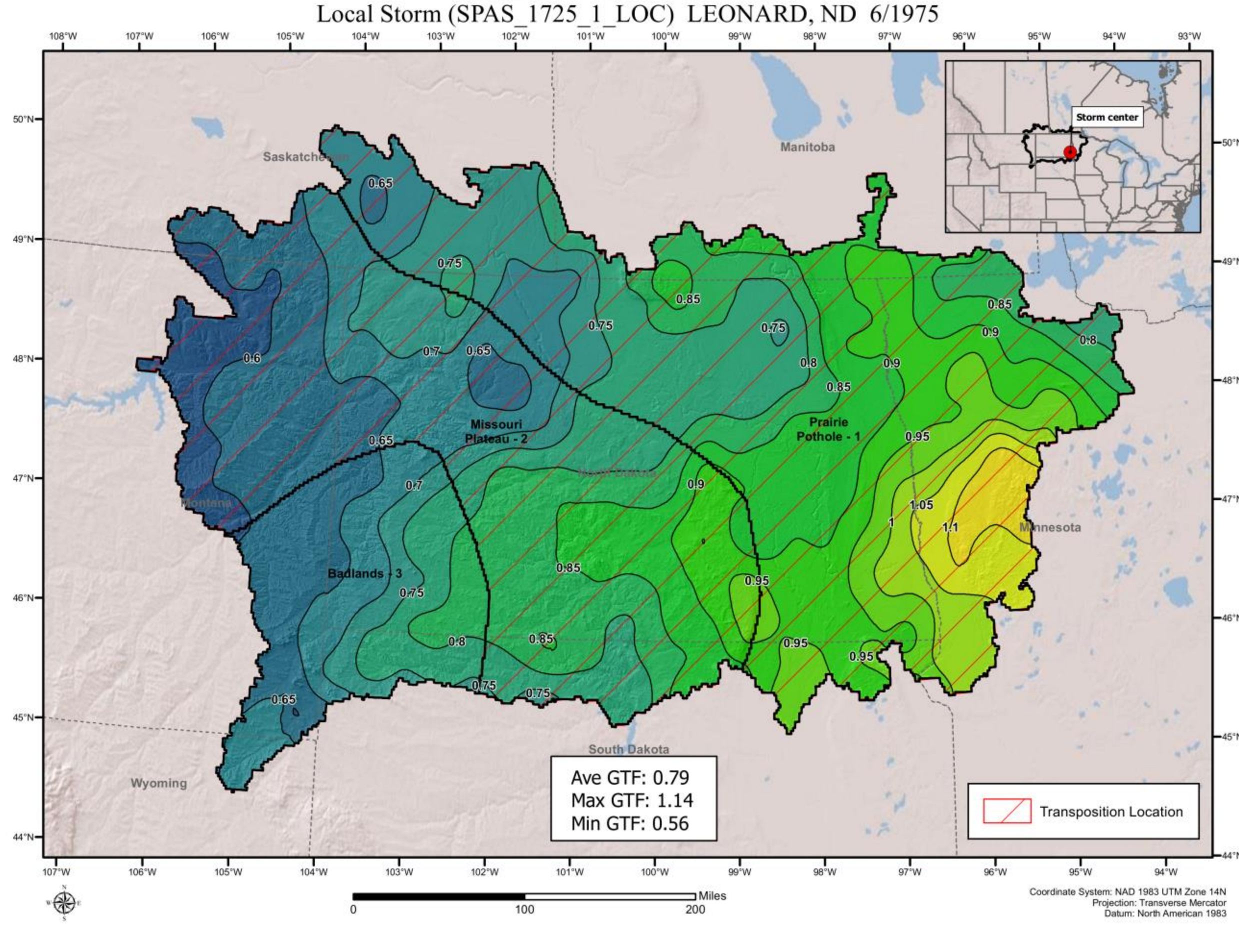
Geographic Transposition Factor
Local Storm (SPAS_1673_1) HARROW, ONT 7/1989



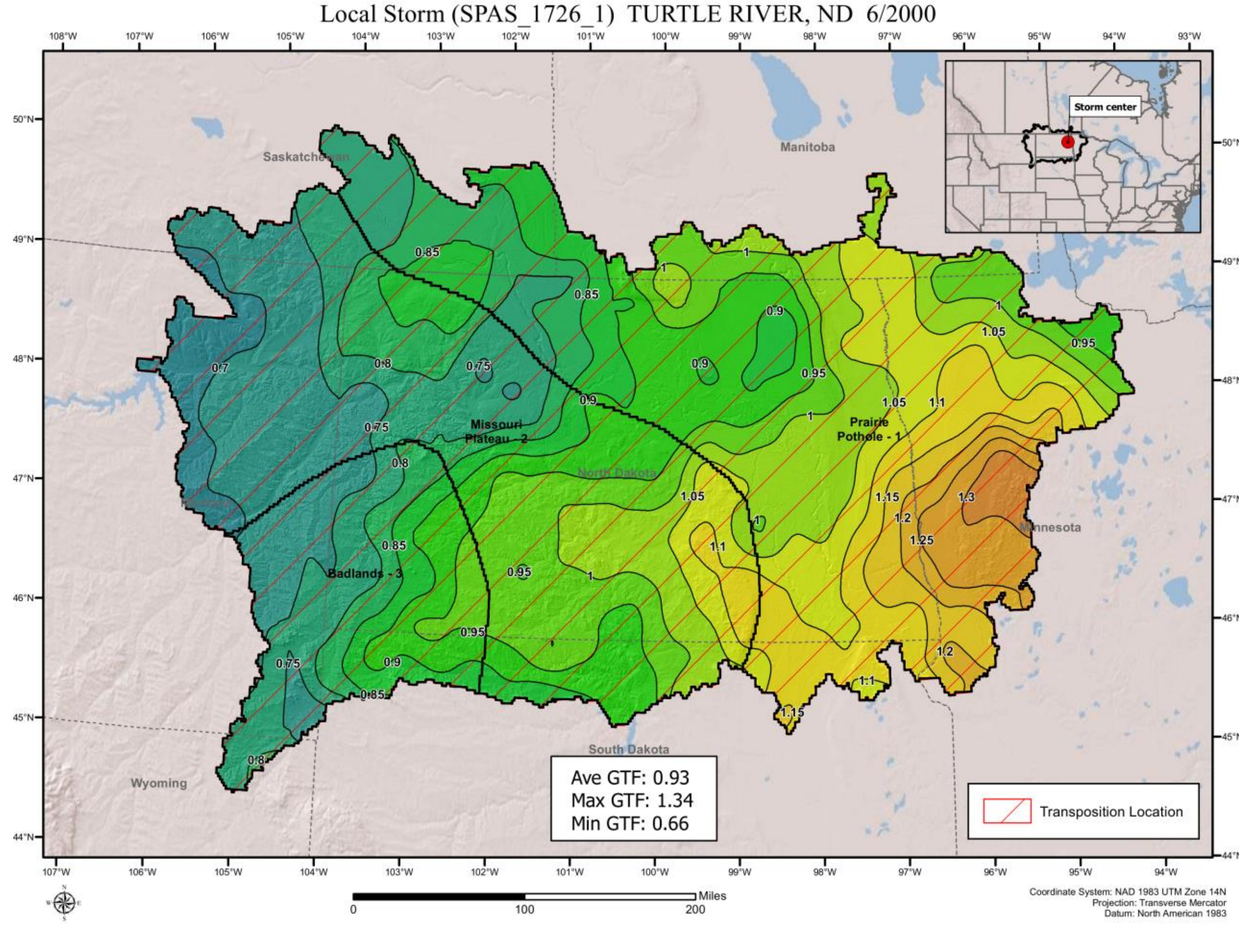
Geographic Transposition Factor
Local Storm (SPAS_1699_1_LOC) HAYWARD, WI 8/1941



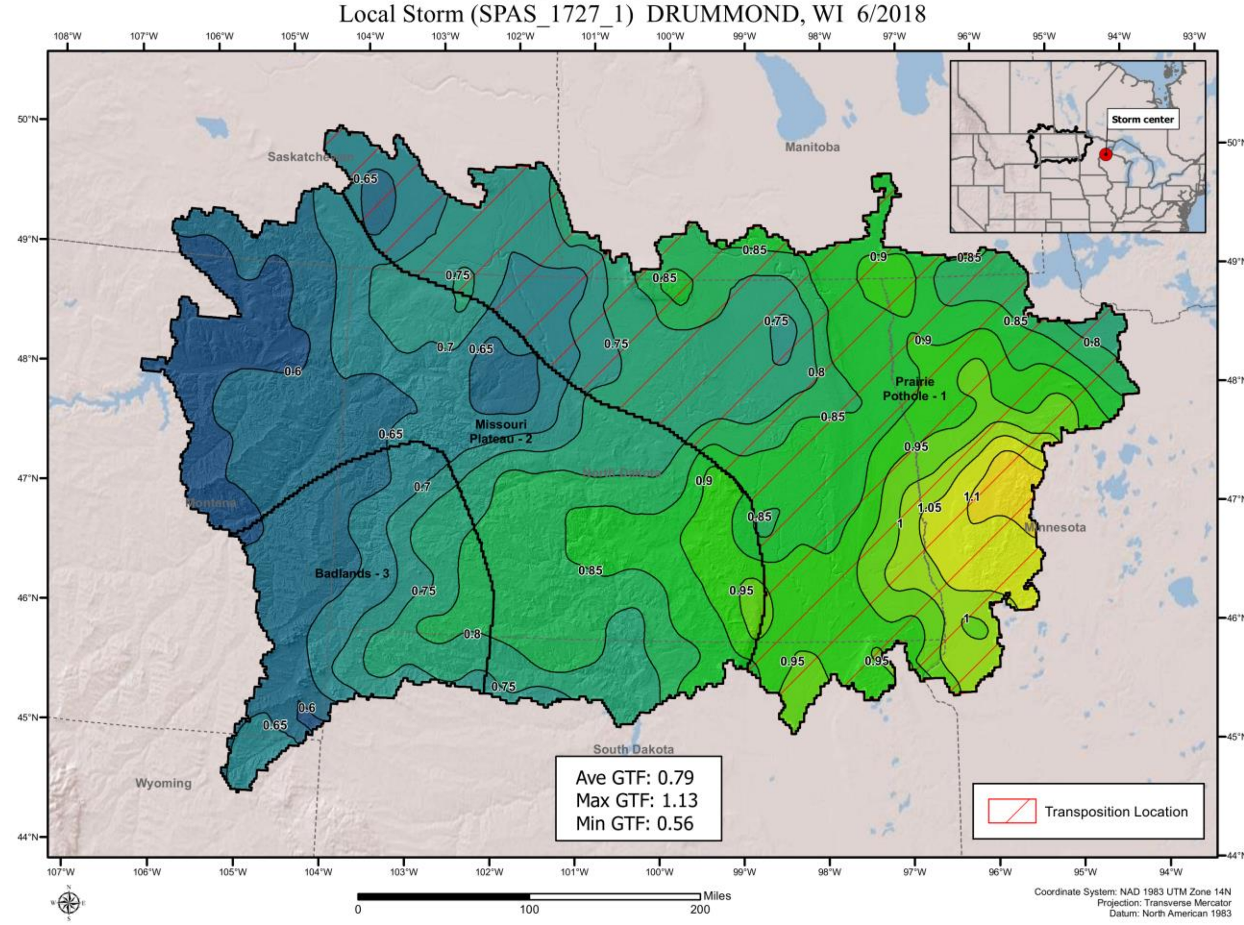
Geographic Transposition Factor
Local Storm (SPAS_1725_1_LOC) LEONARD, ND 6/1975



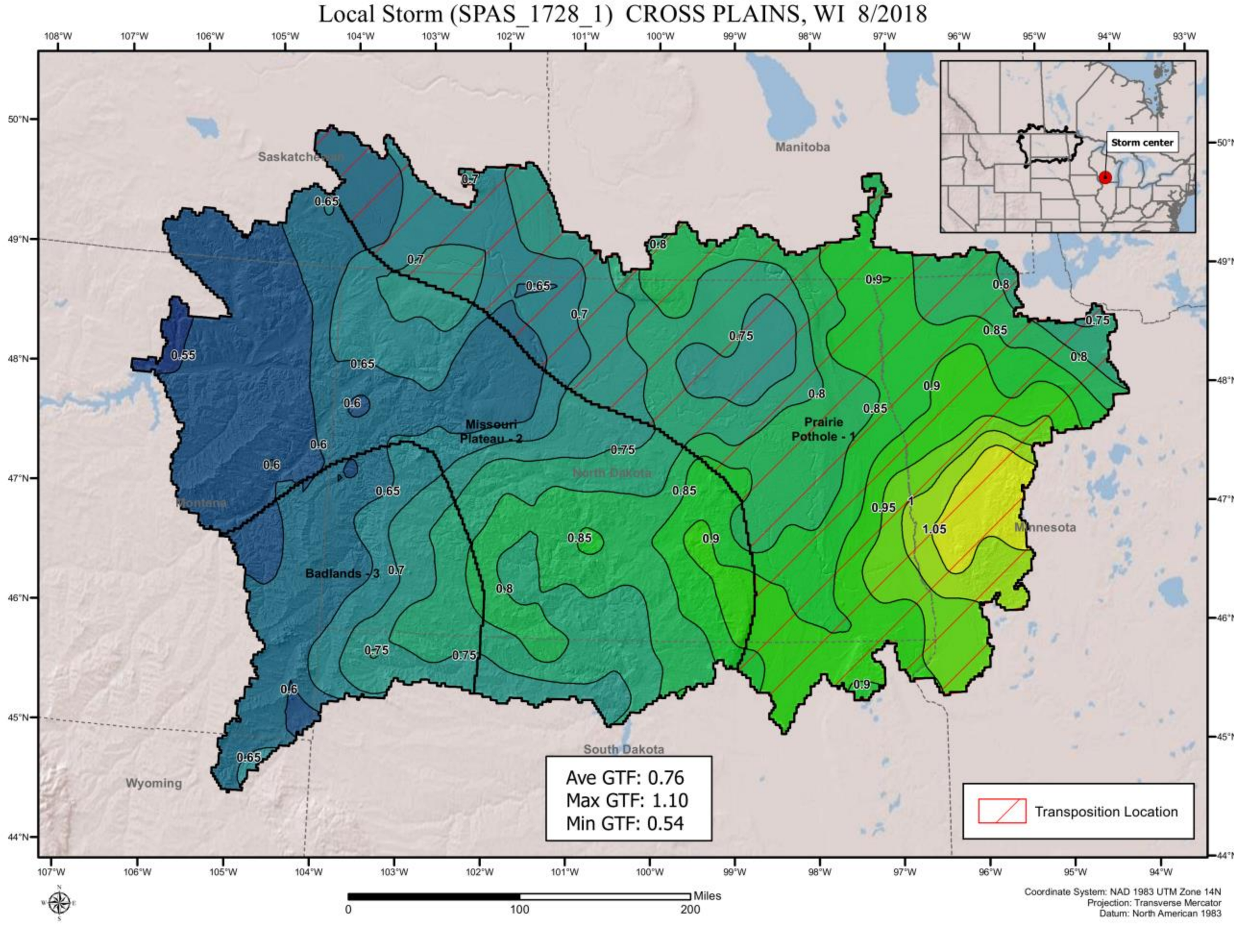
Geographic Transposition Factor
Local Storm (SPAS_1726_1) TURTLE RIVER, ND 6/2000



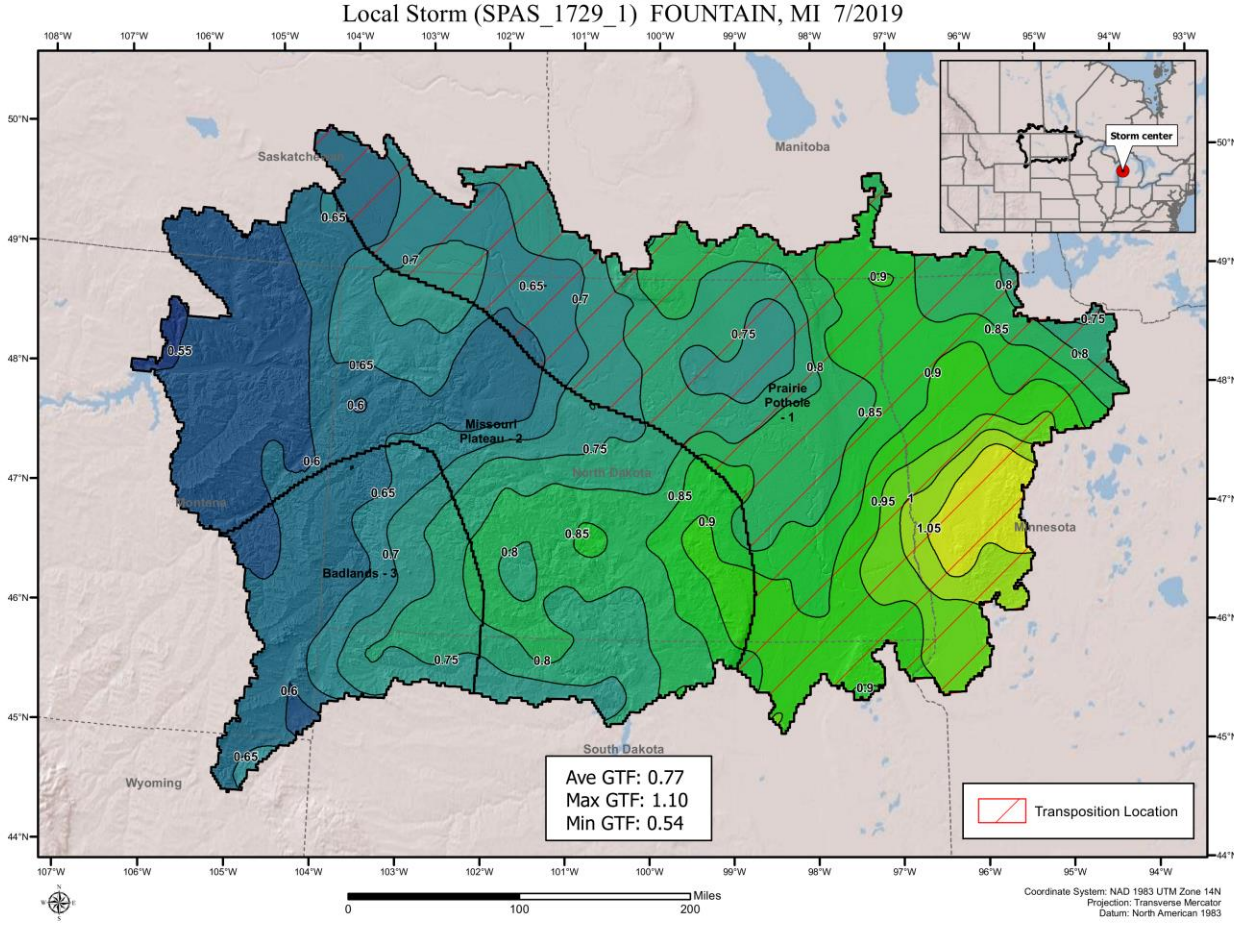
Geographic Transposition Factor
Local Storm (SPAS_1727_1) DRUMMOND, WI 6/2018



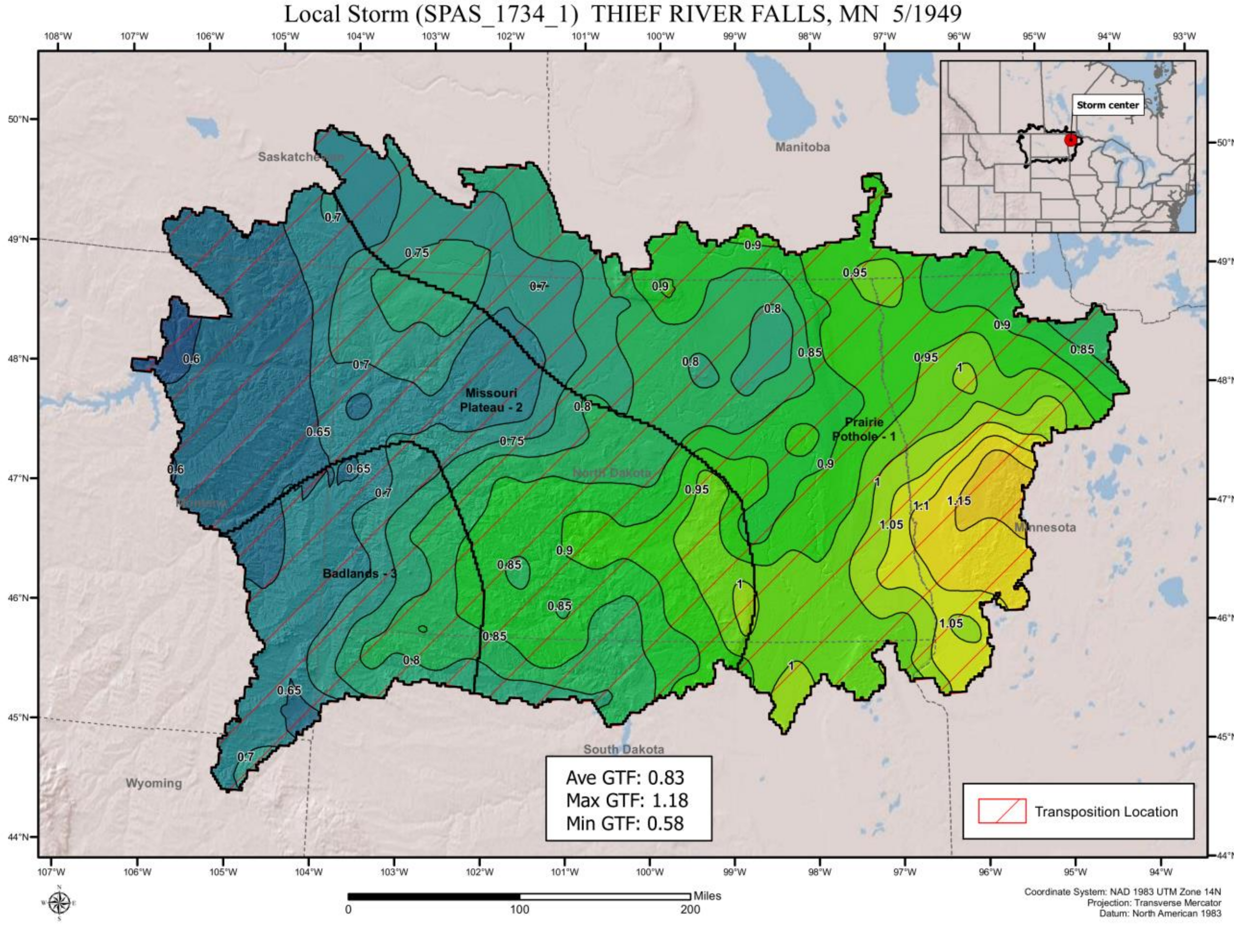
Geographic Transposition Factor
Local Storm (SPAS_1728_1) CROSS PLAINS, WI 8/2018



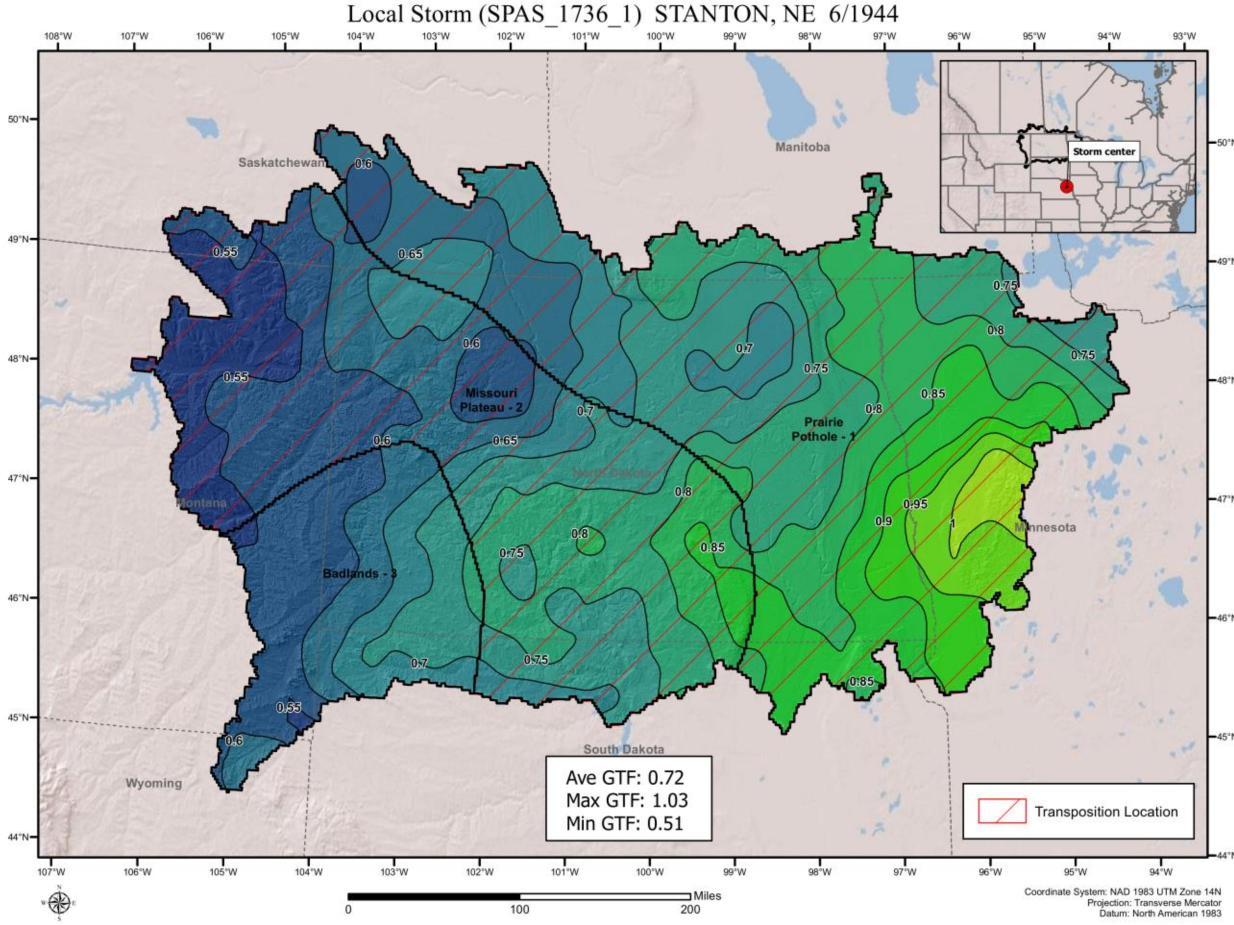
Geographic Transposition Factor
Local Storm (SPAS_1729_1) FOUNTAIN, MI 7/2019



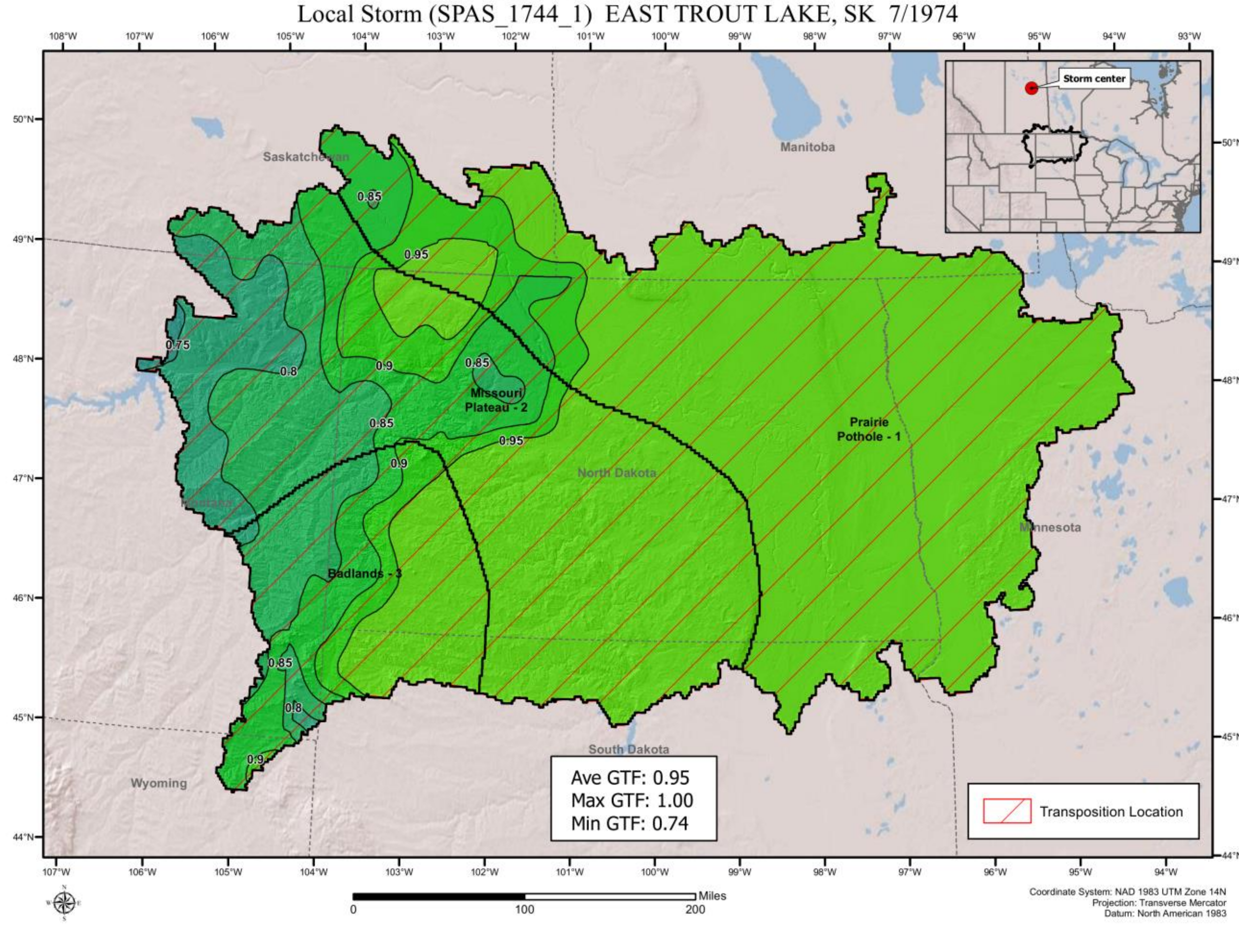
Geographic Transposition Factor
Local Storm (SPAS_1734_1) THIEF RIVER FALLS, MN 5/1949



Geographic Transposition Factor
Local Storm (SPAS_1736_1) STANTON, NE 6/1944

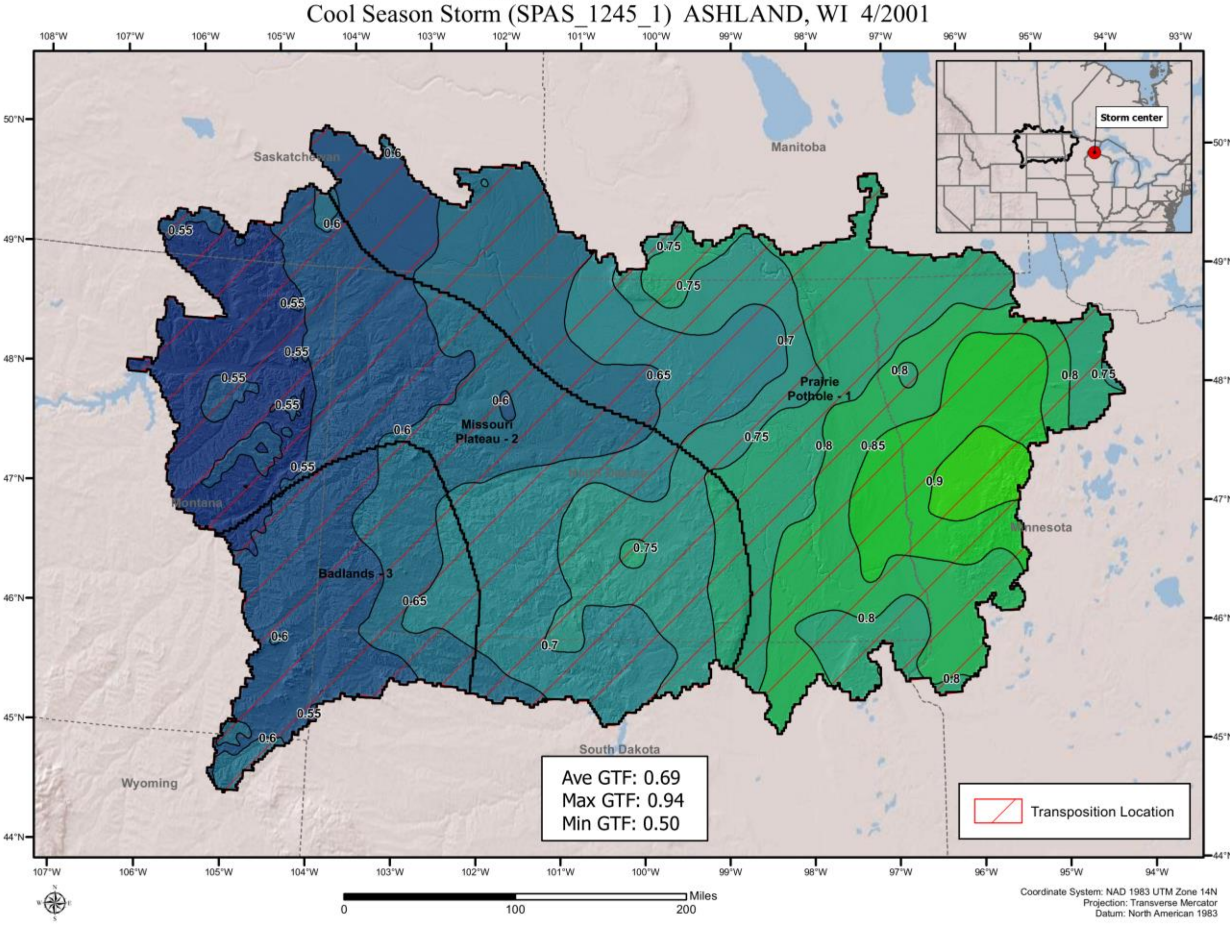


Geographic Transposition Factor
Local Storm (SPAS_1744_1) EAST TROUT LAKE, SK 7/1974

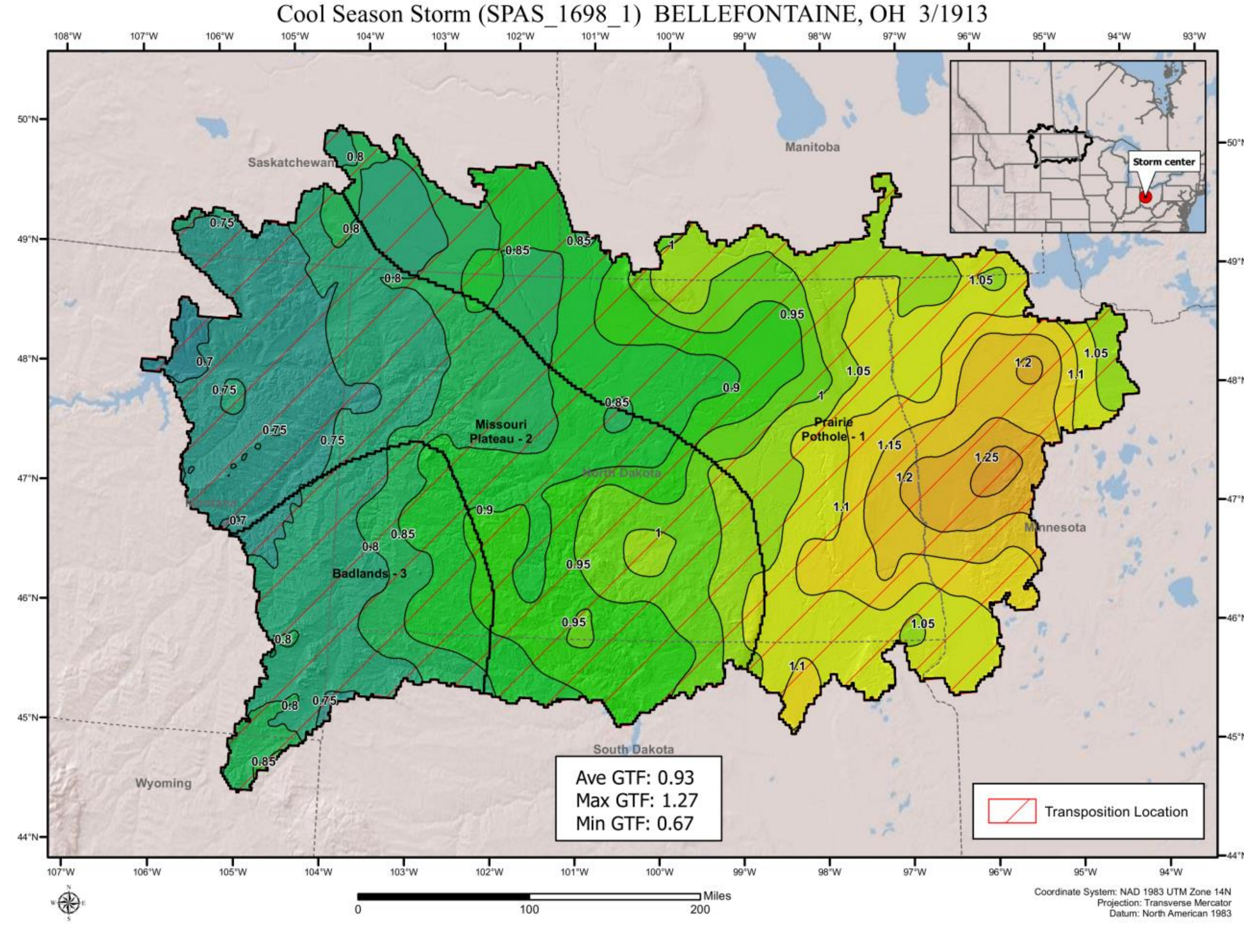


Cool Season Storms

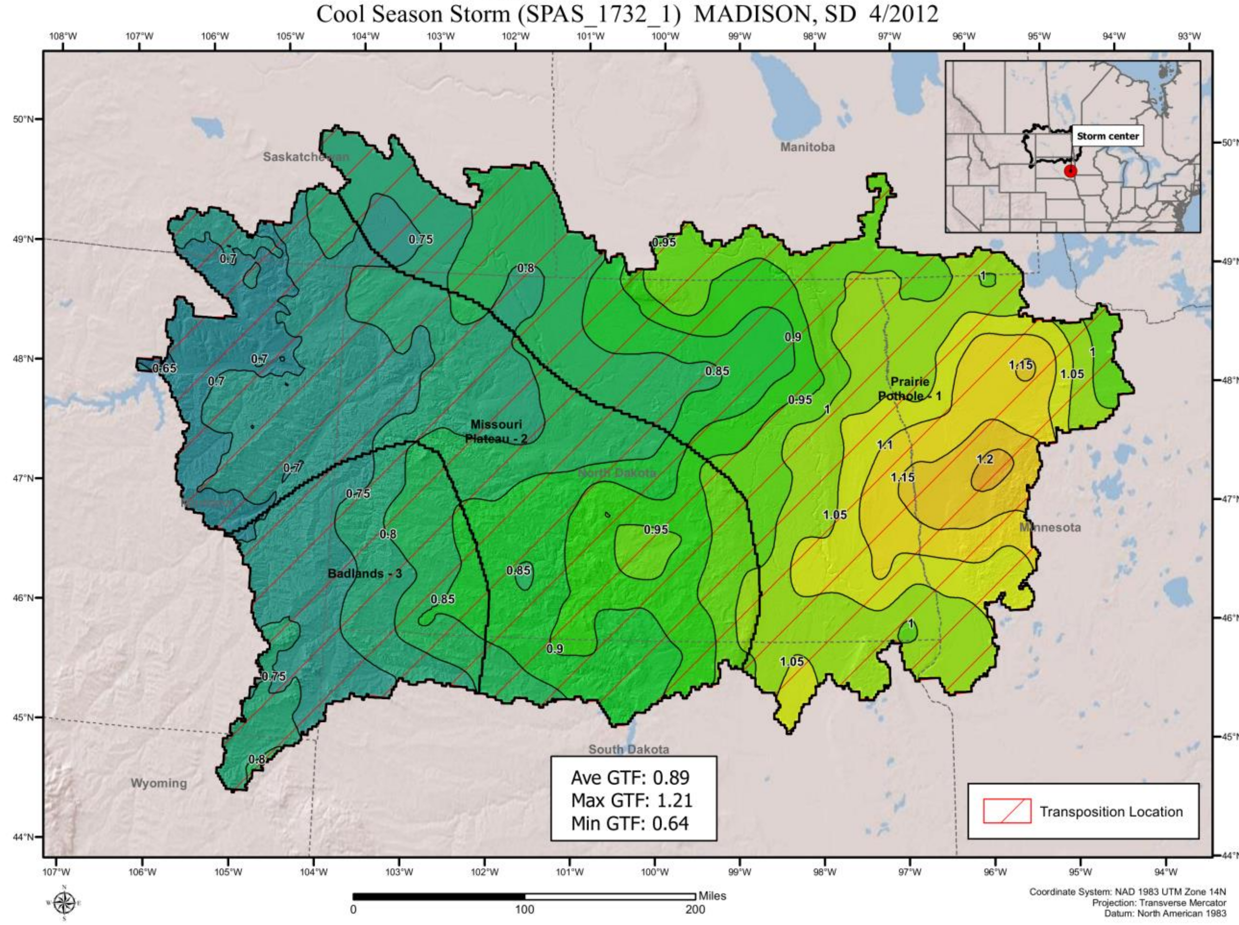
Geographic Transposition Factor
Cool Season Storm (SPAS_1245_1) ASHLAND, WI 4/2001



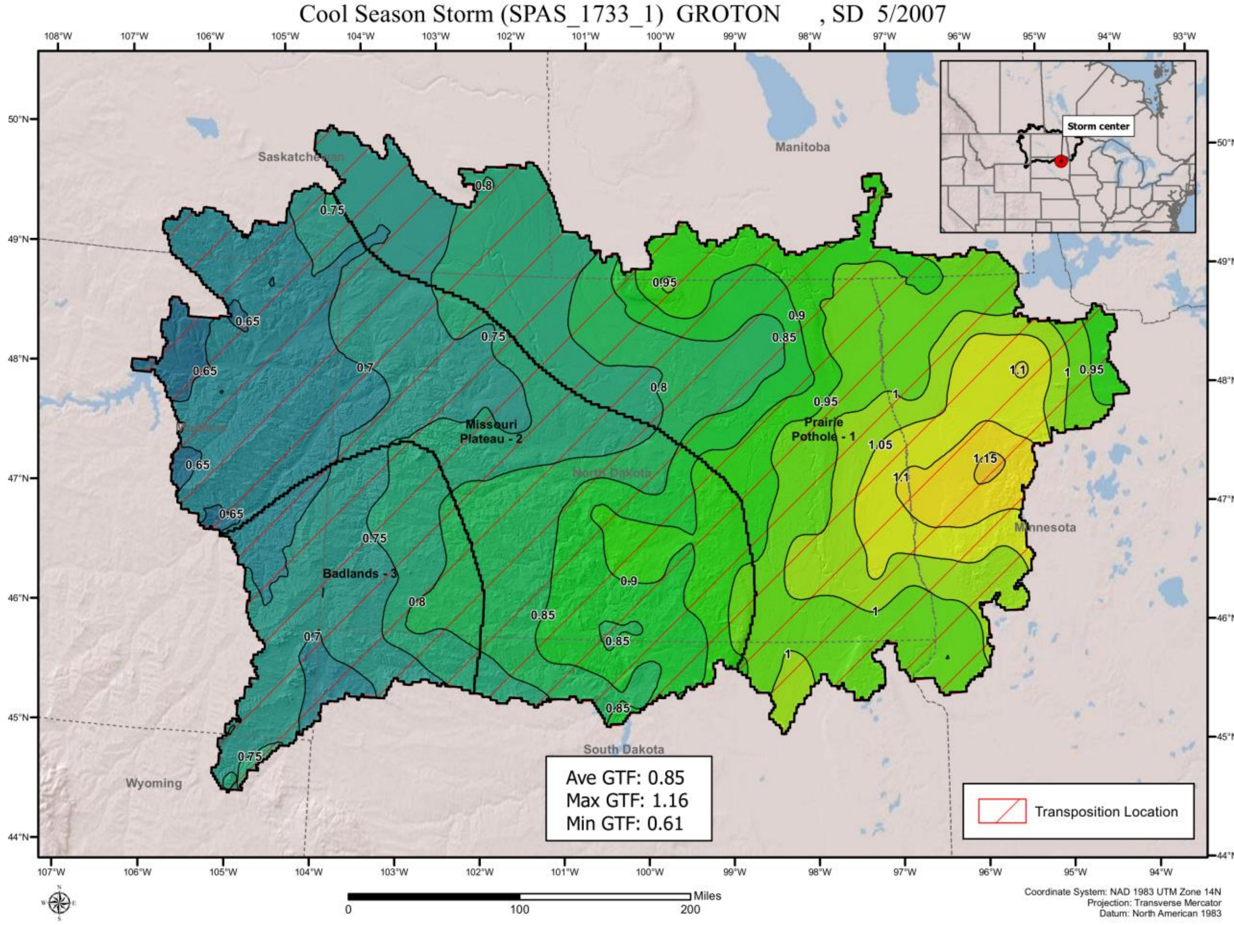
Geographic Transposition Factor
Cool Season Storm (SPAS_1698_1) BELLEFONTAINE, OH 3/1913



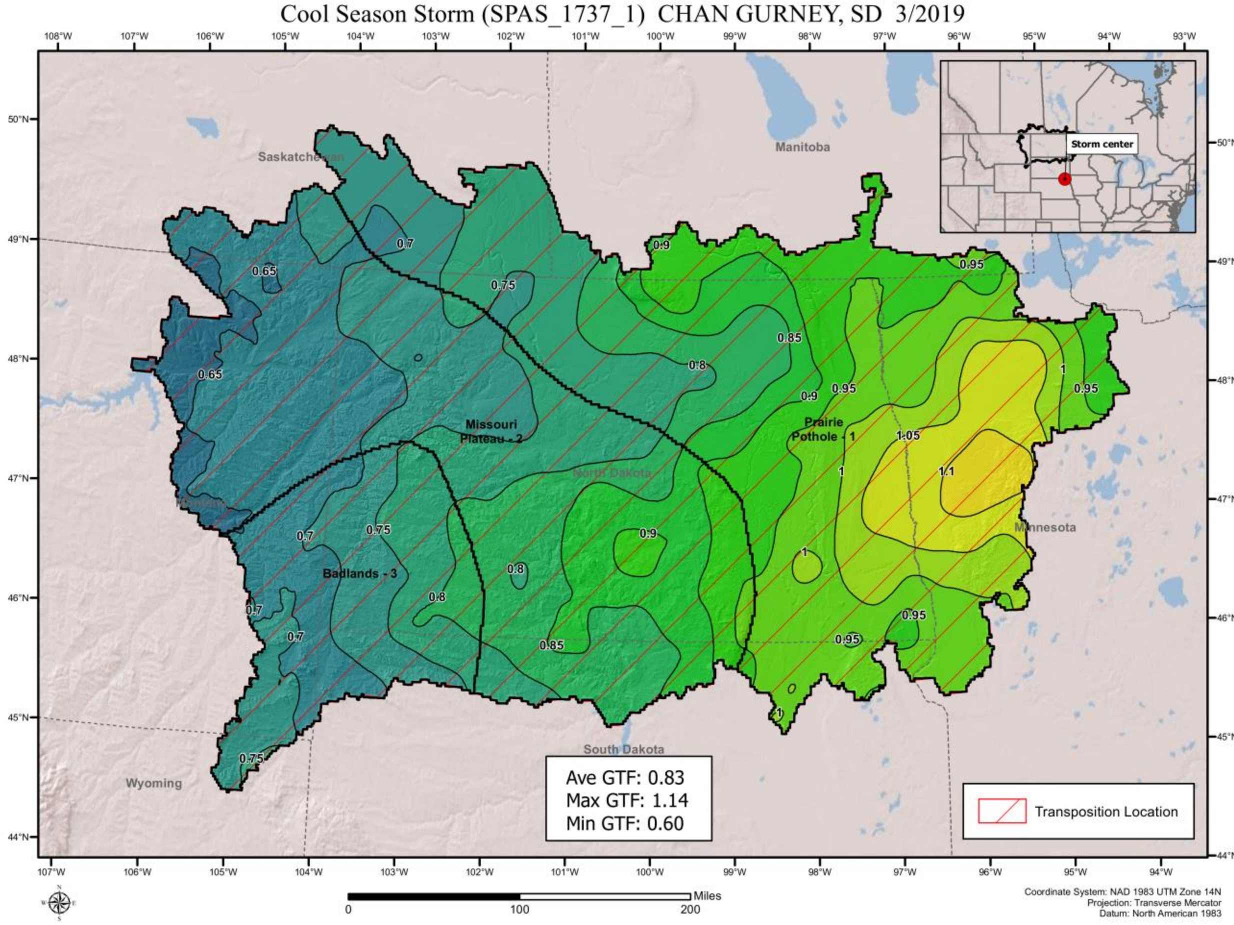
Geographic Transposition Factor
Cool Season Storm (SPAS_1732_1) MADISON, SD 4/2012



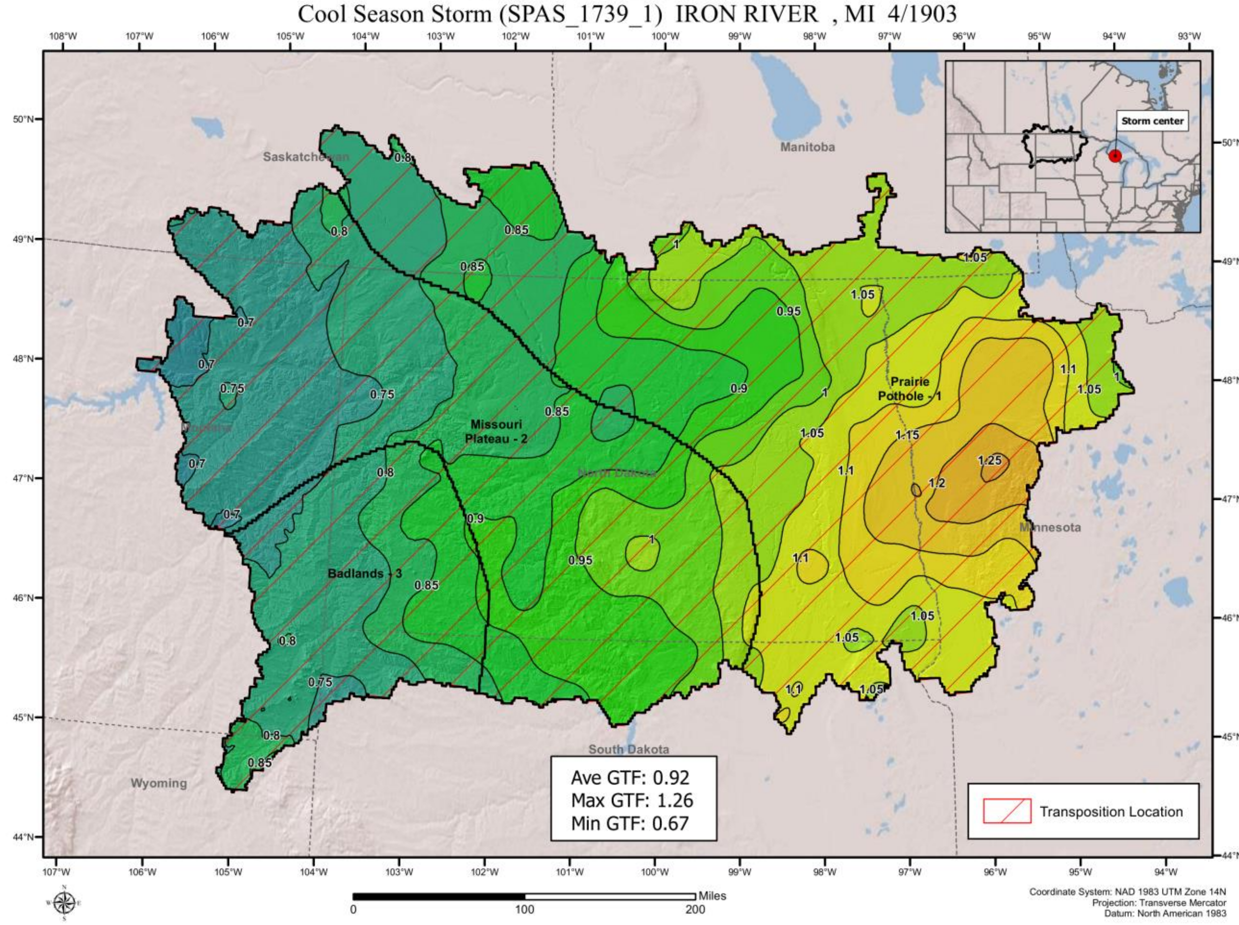
Geographic Transposition Factor
Cool Season Storm (SPAS_1733_1) GROTON , SD 5/2007



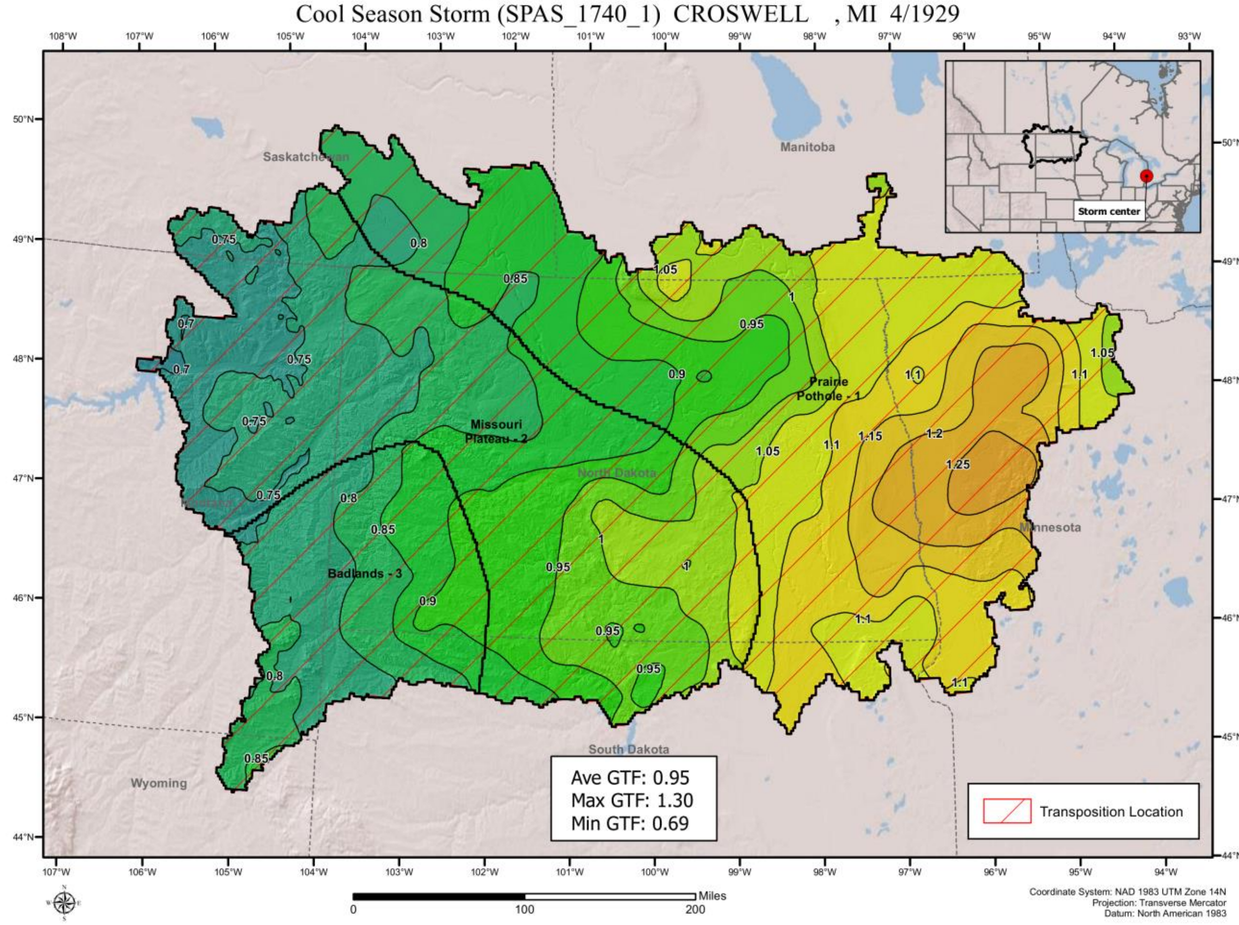
Geographic Transposition Factor
Cool Season Storm (SPAS_1737_1) CHAN GURNEY, SD 3/2019



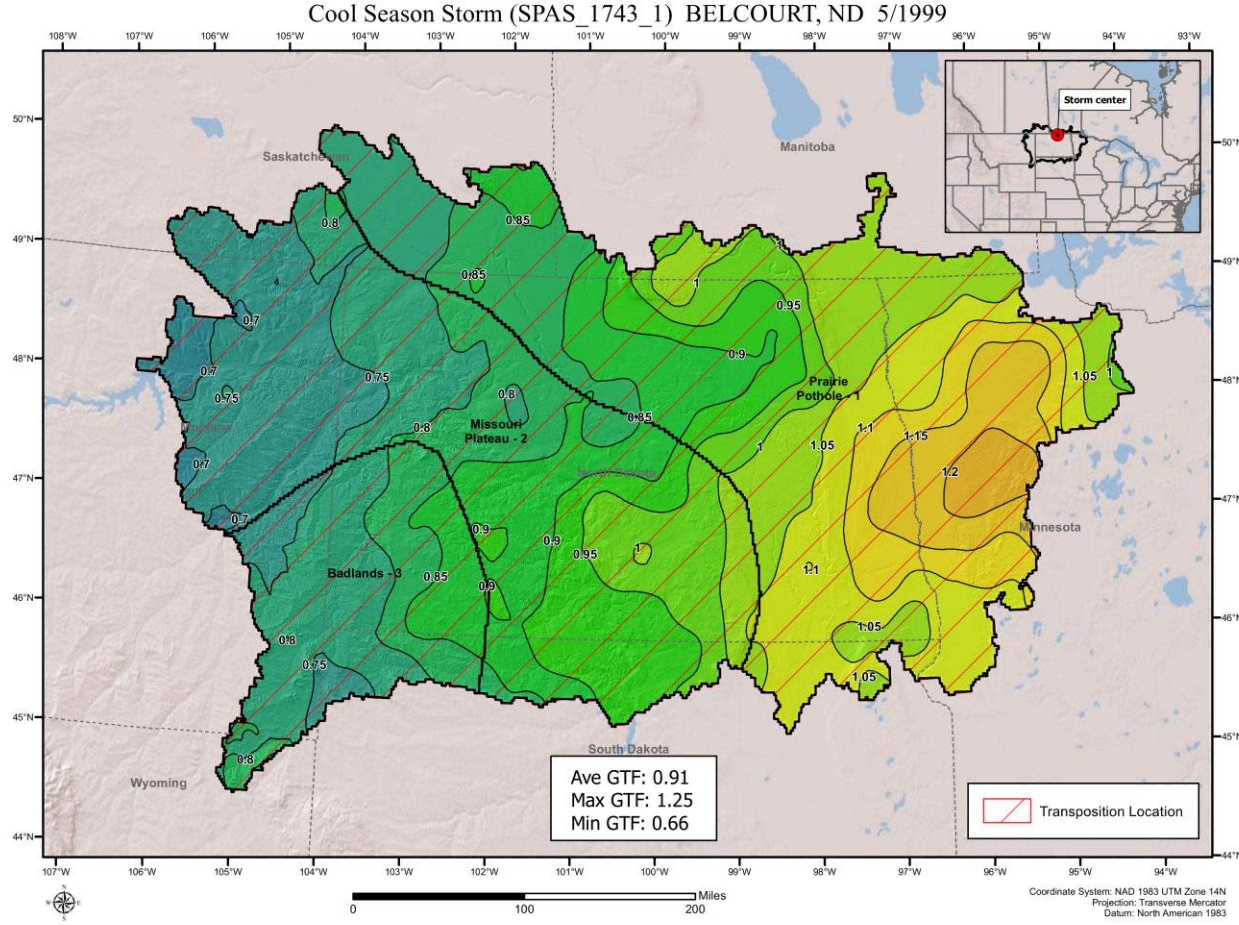
Geographic Transposition Factor
Cool Season Storm (SPAS_1739_1) IRON RIVER , MI 4/1903



Geographic Transposition Factor
Cool Season Storm (SPAS_1740_1) CROSWELL , MI 4/1929



Geographic Transposition Factor
Cool Season Storm (SPAS_1743_1) BELCOURT, ND 5/1999

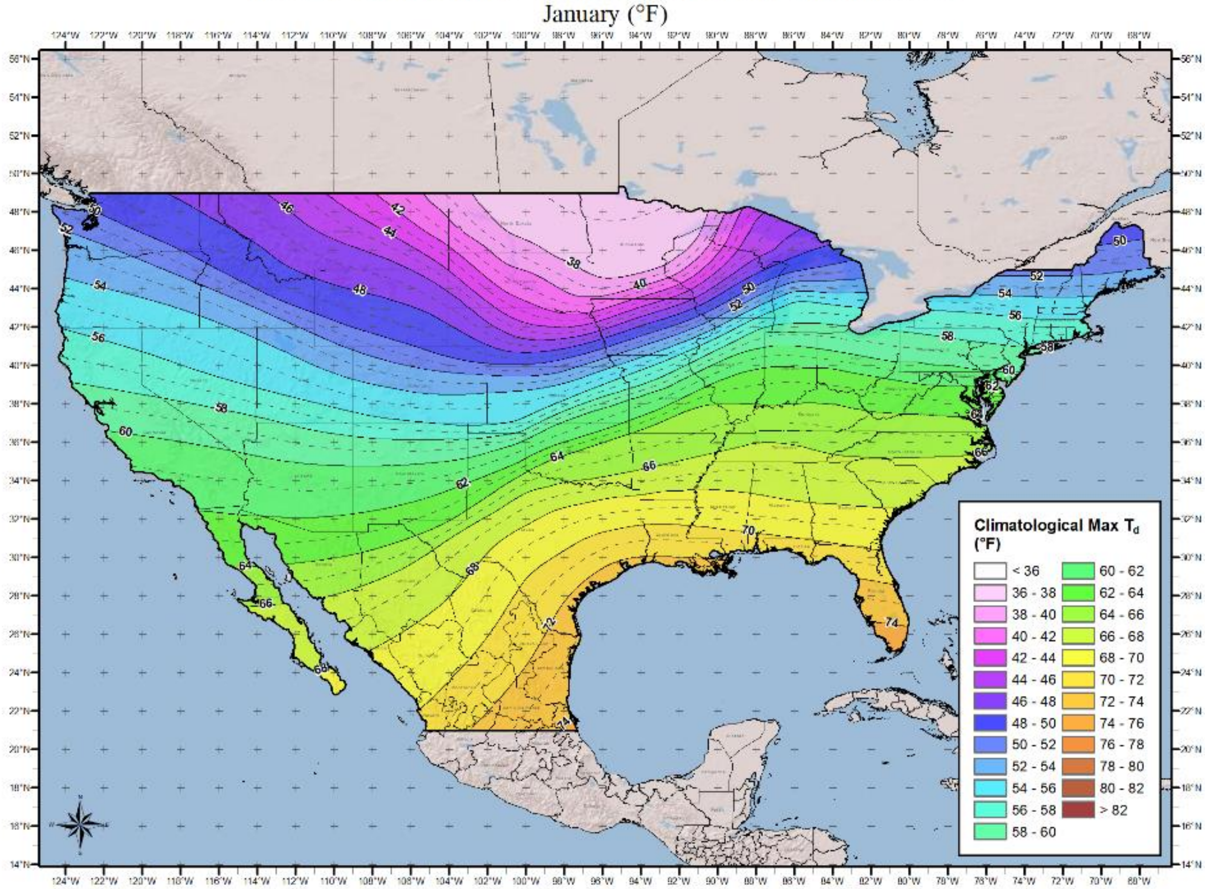


Appendix C

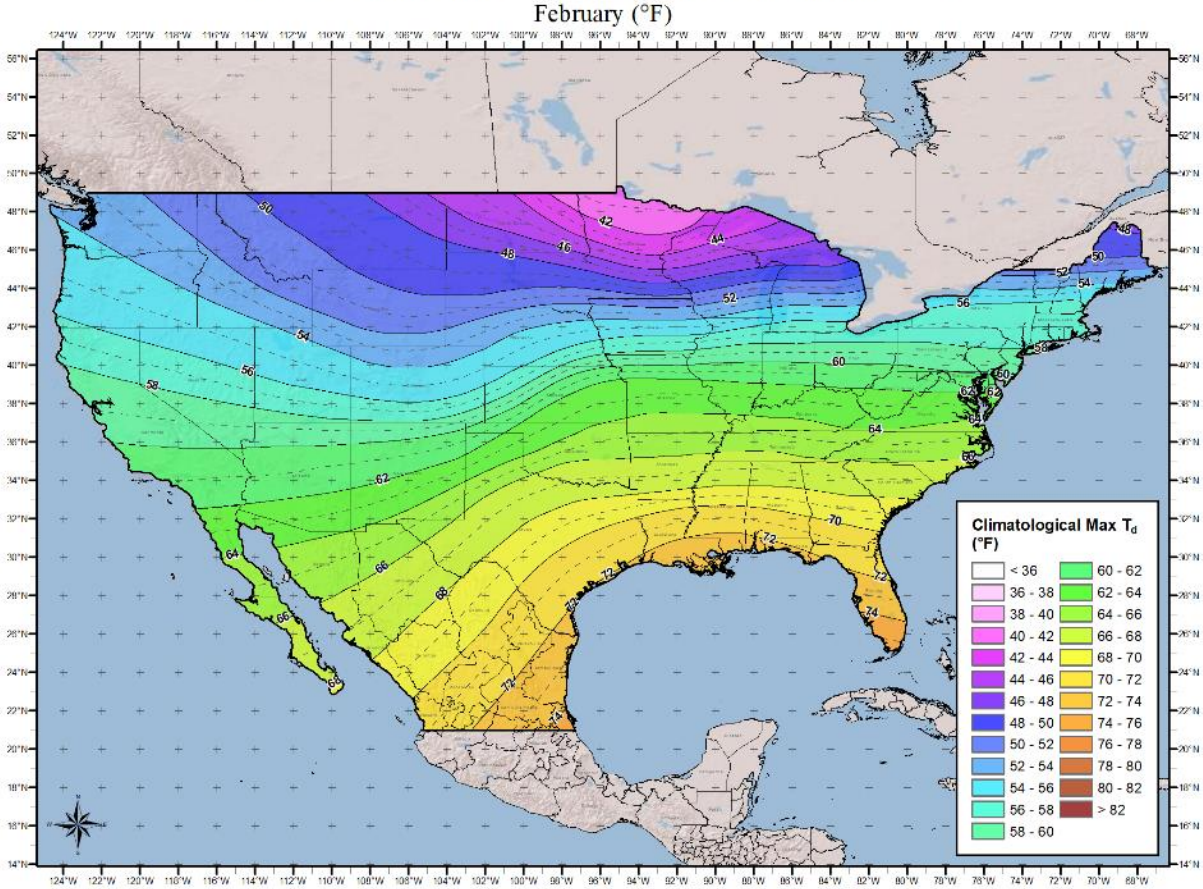
100-year Return Frequency Maximum Average Dew Point Temperature Climatology Maps

6-hour 1000mb Dew Point Maps

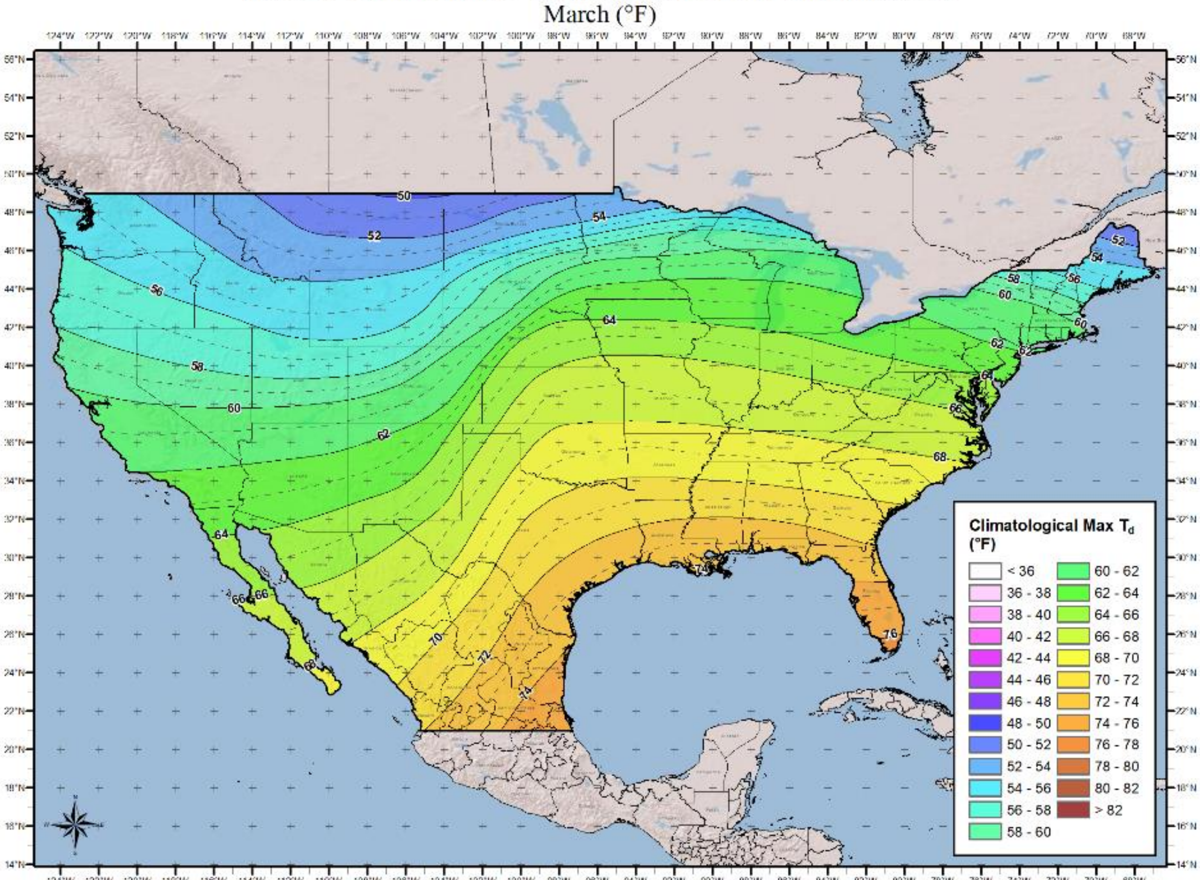
100-year Return Frequency 6-hour Maximum Dew Point Climatology
January (°F)



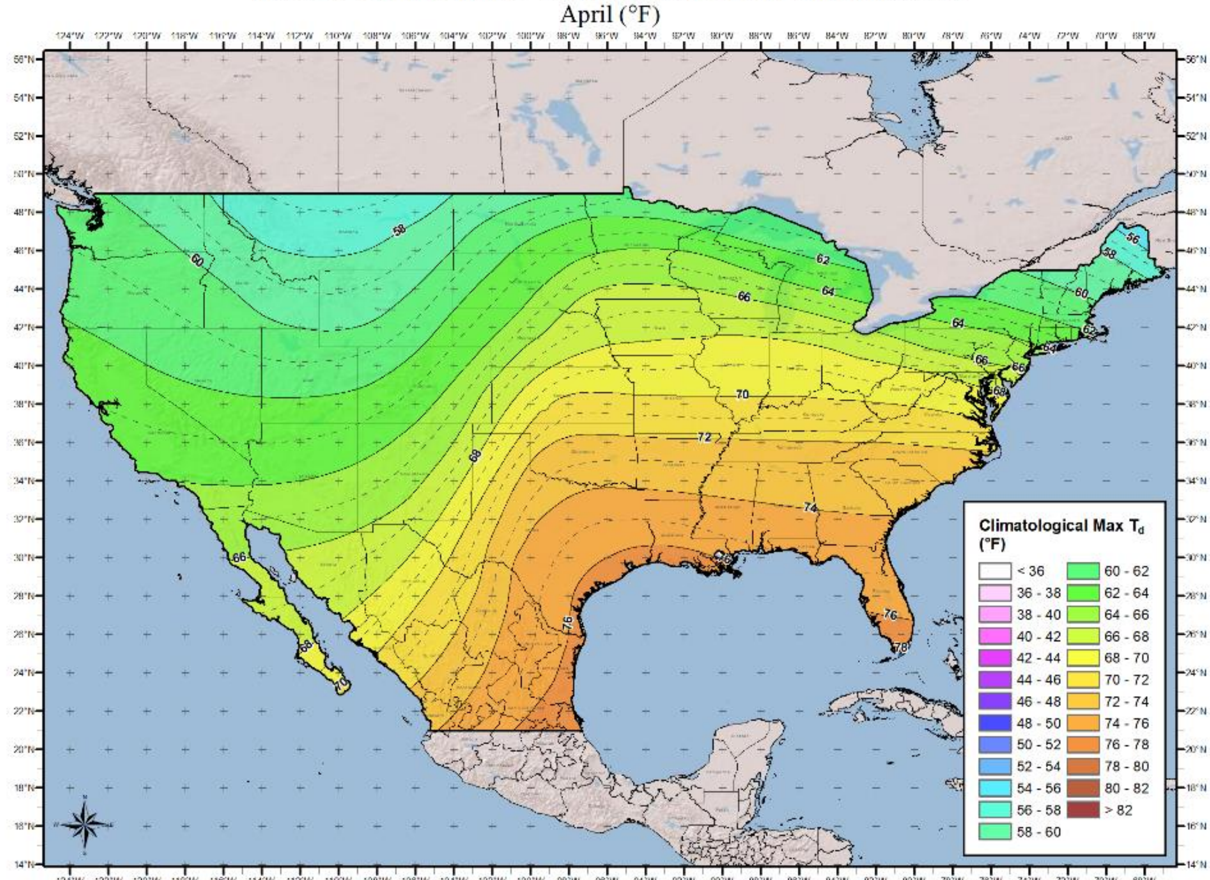
100-year Return Frequency 6-hour Maximum Dew Point Climatology
February (°F)



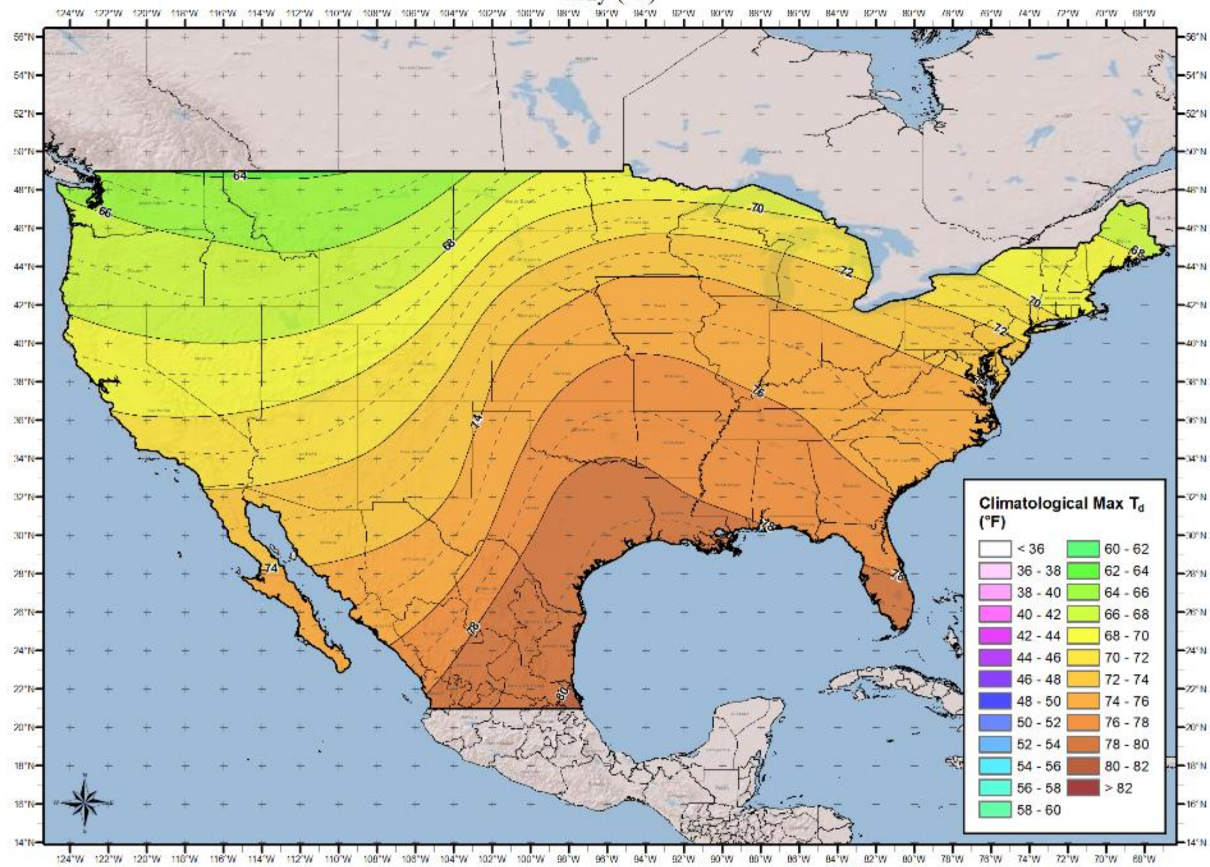
100-year Return Frequency 6-hour Maximum Dew Point Climatology
March (°F)



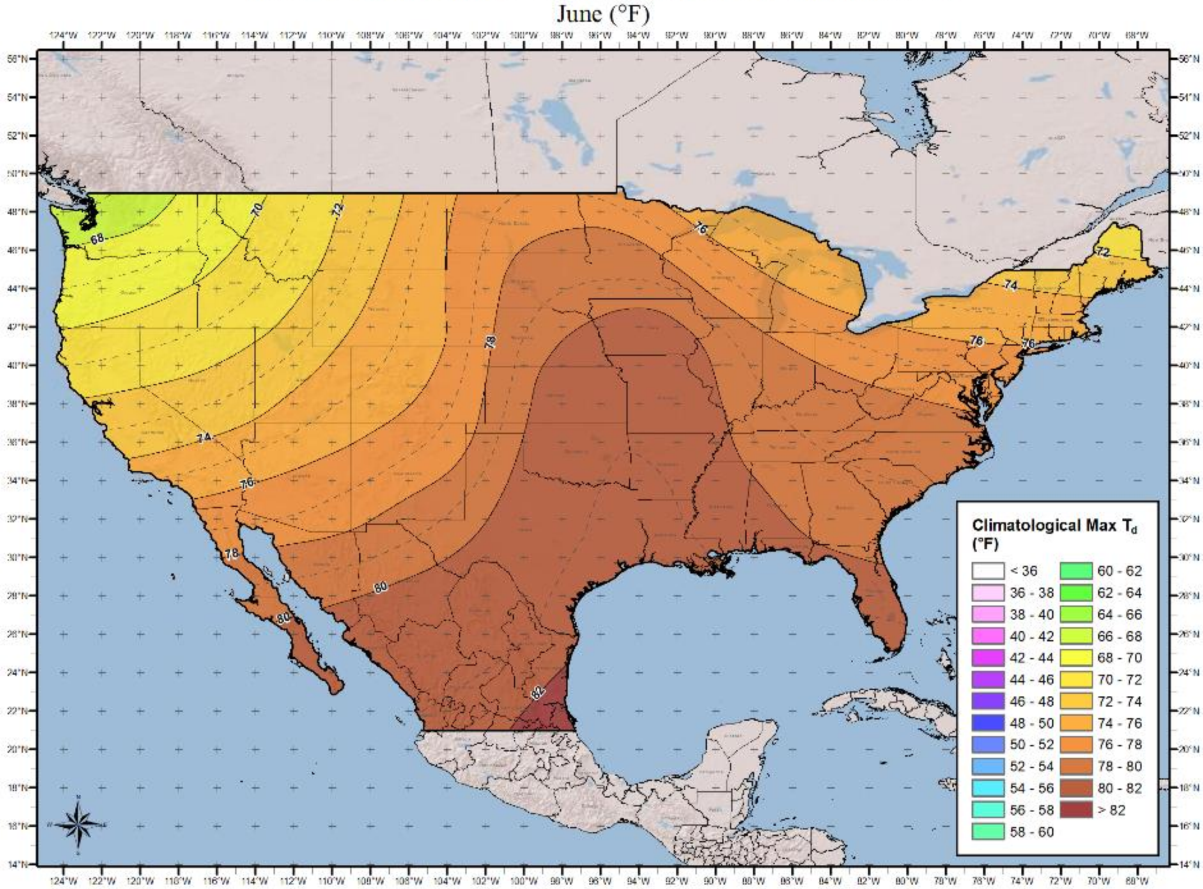
100-year Return Frequency 6-hour Maximum Dew Point Climatology
April (°F)



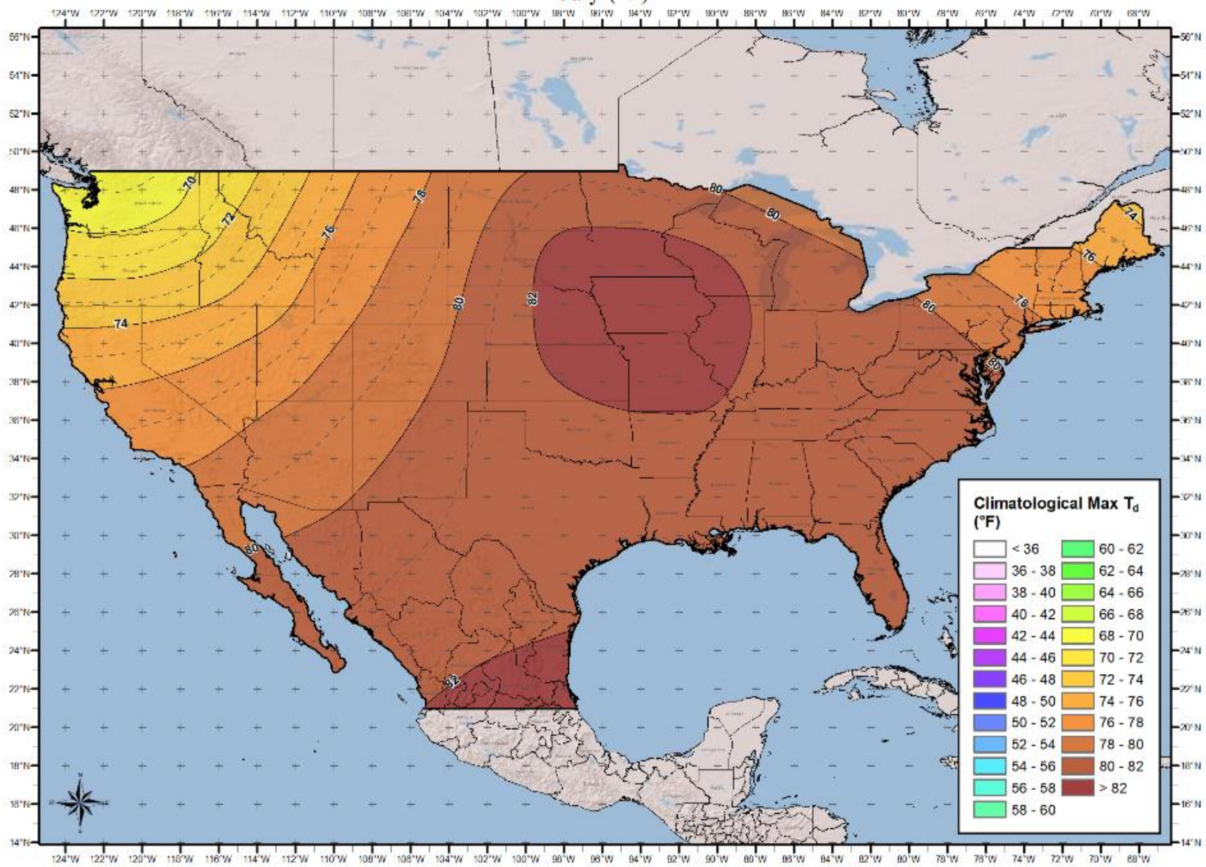
100-year Return Frequency 6-hour Maximum Dew Point Climatology
May (°F)



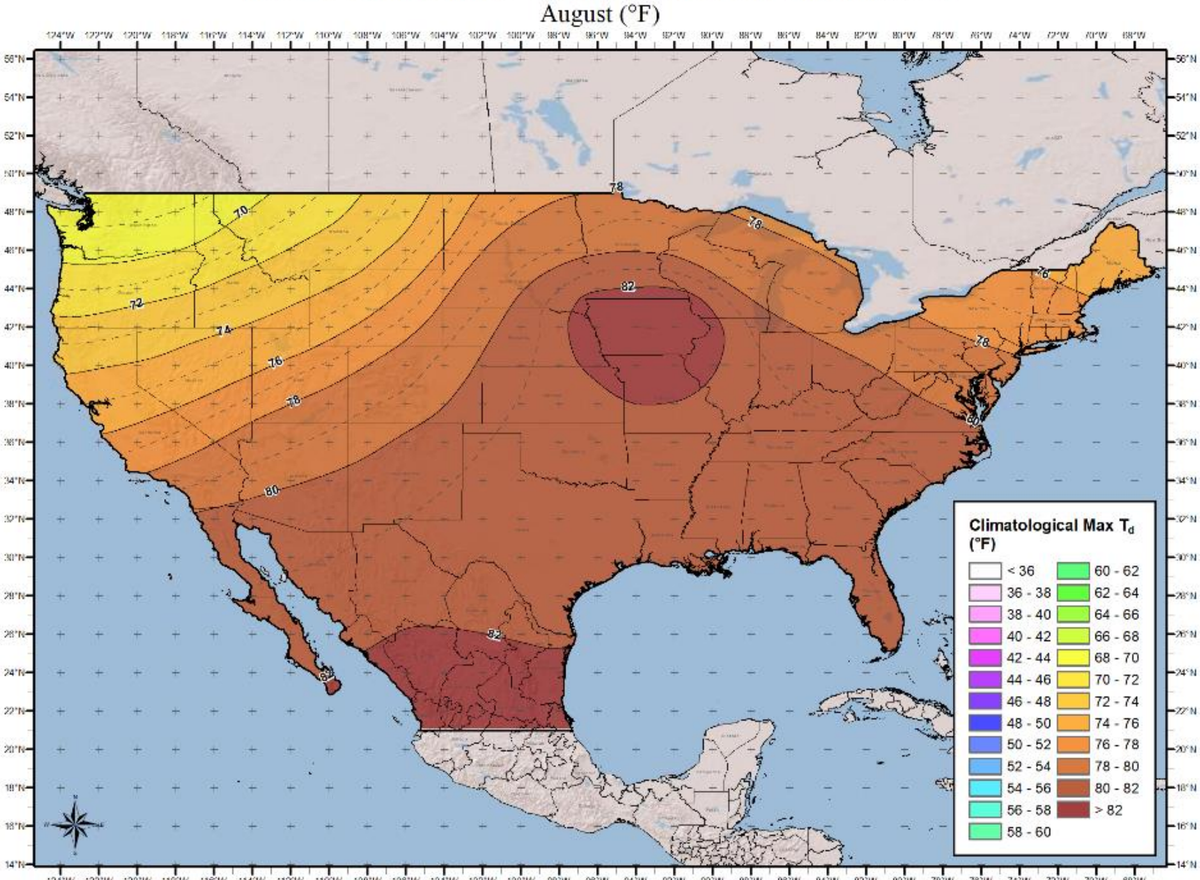
100-year Return Frequency 6-hour Maximum Dew Point Climatology
June (°F)



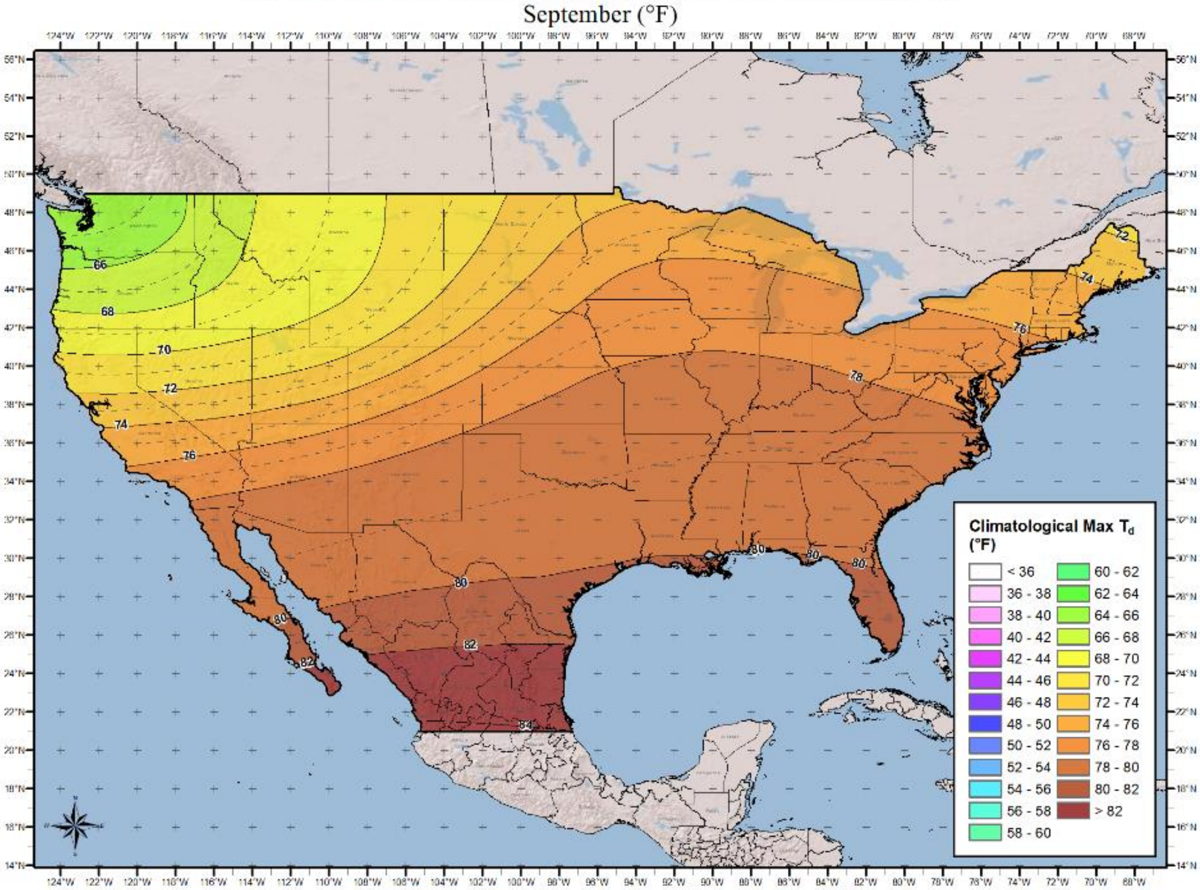
100-year Return Frequency 6-hour Maximum Dew Point Climatology
July (°F)



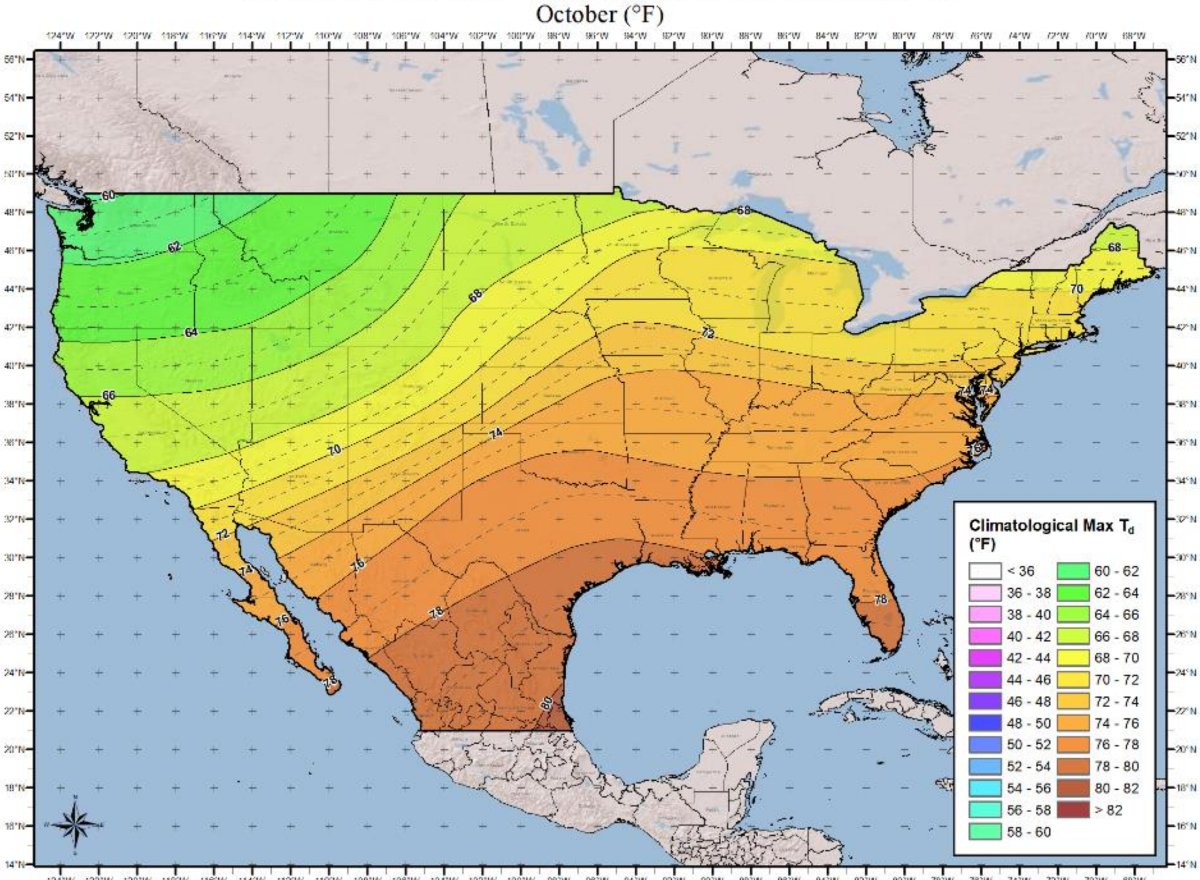
100-year Return Frequency 6-hour Maximum Dew Point Climatology
August (°F)



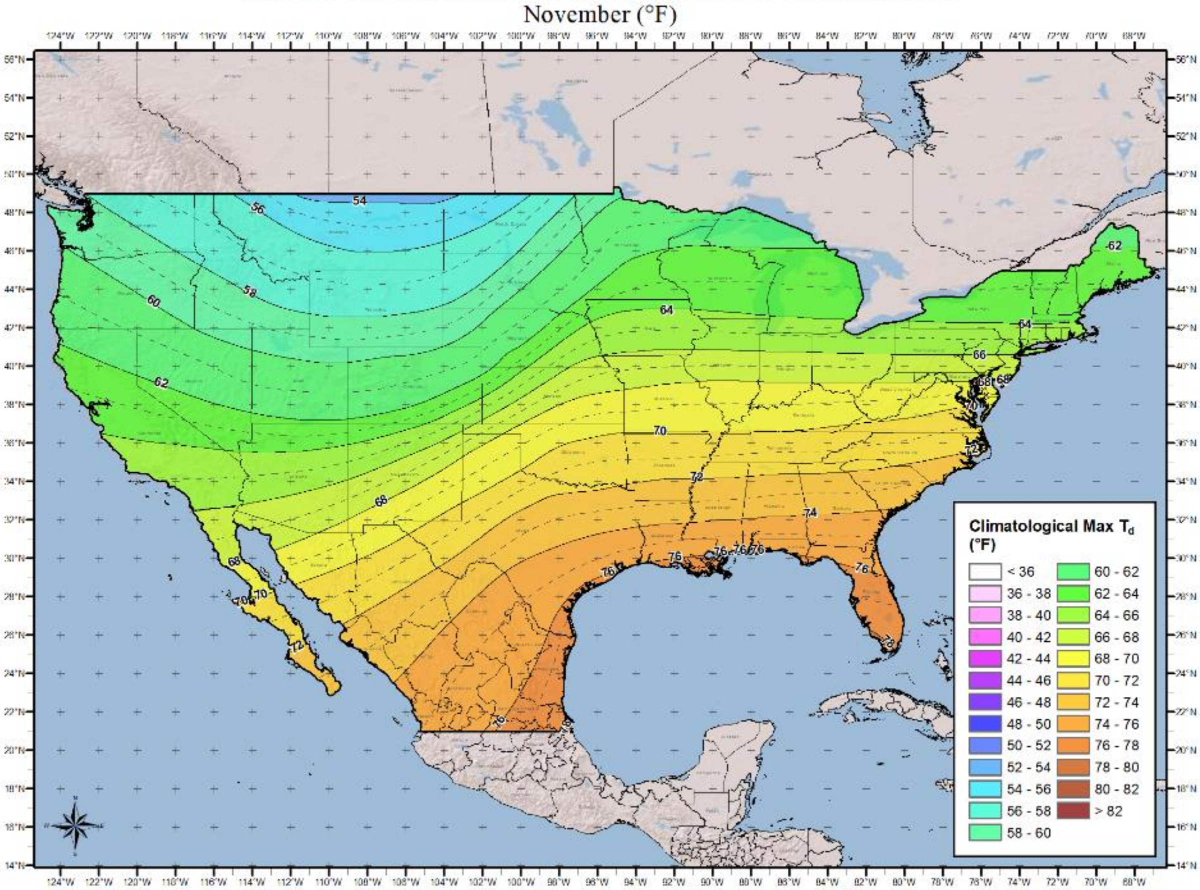
100-year Return Frequency 6-hour Maximum Dew Point Climatology
September (°F)



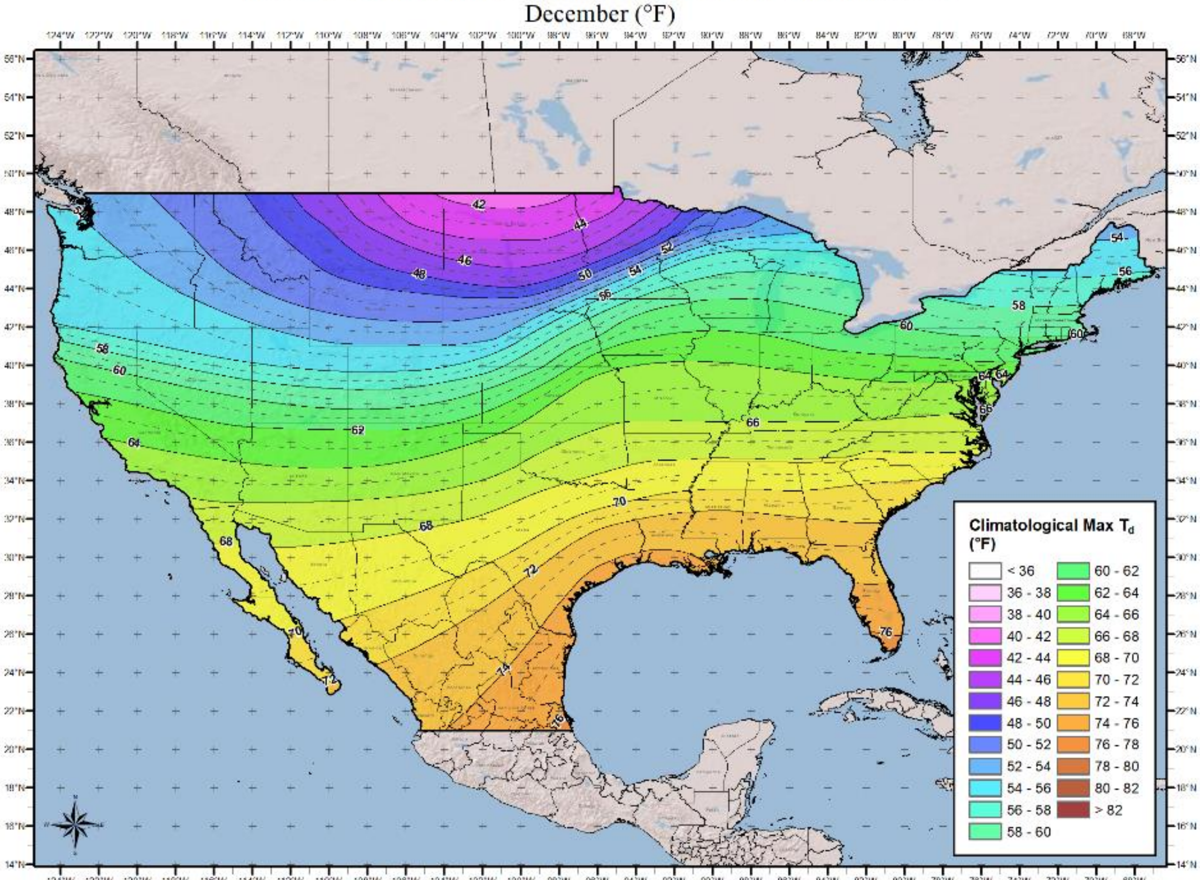
100-year Return Frequency 6-hour Maximum Dew Point Climatology
October (°F)



100-year Return Frequency 6-hour Maximum Dew Point Climatology
November (°F)

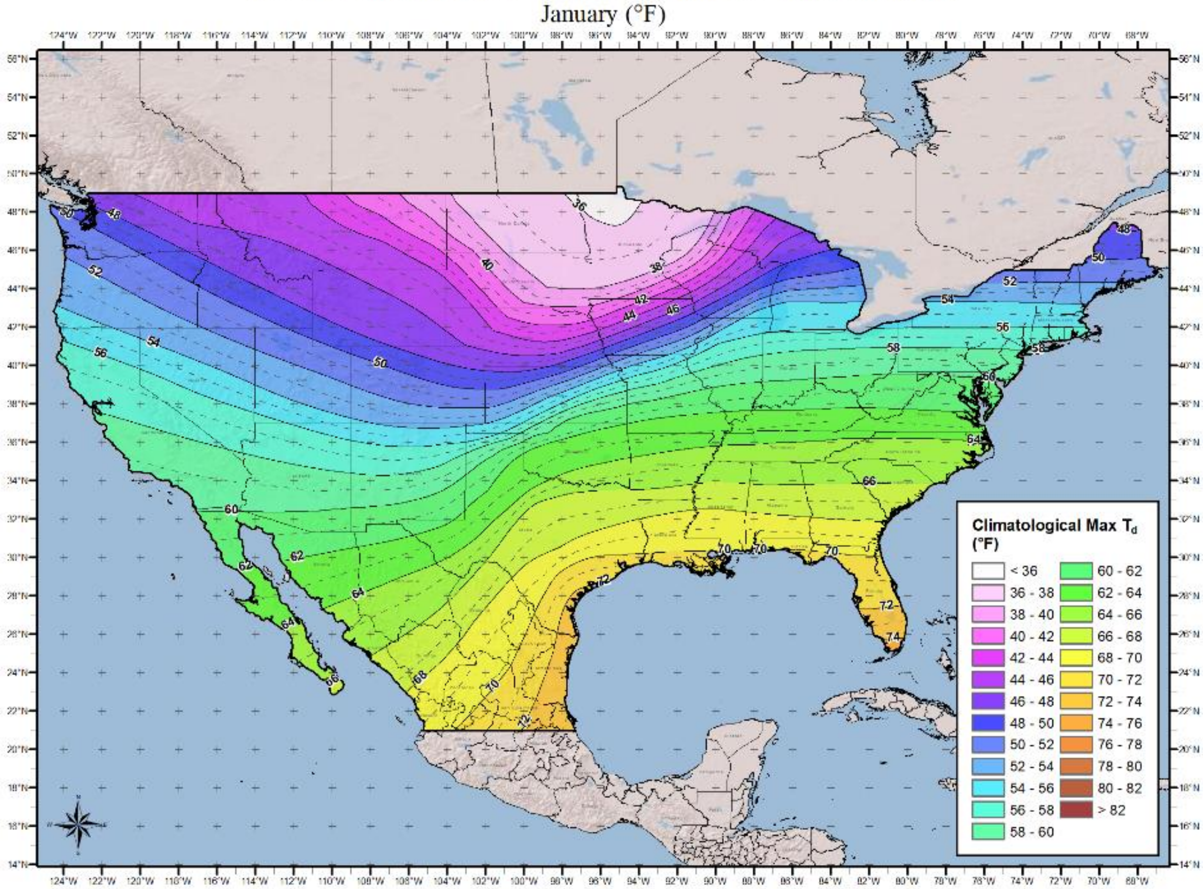


100-year Return Frequency 6-hour Maximum Dew Point Climatology
December (°F)

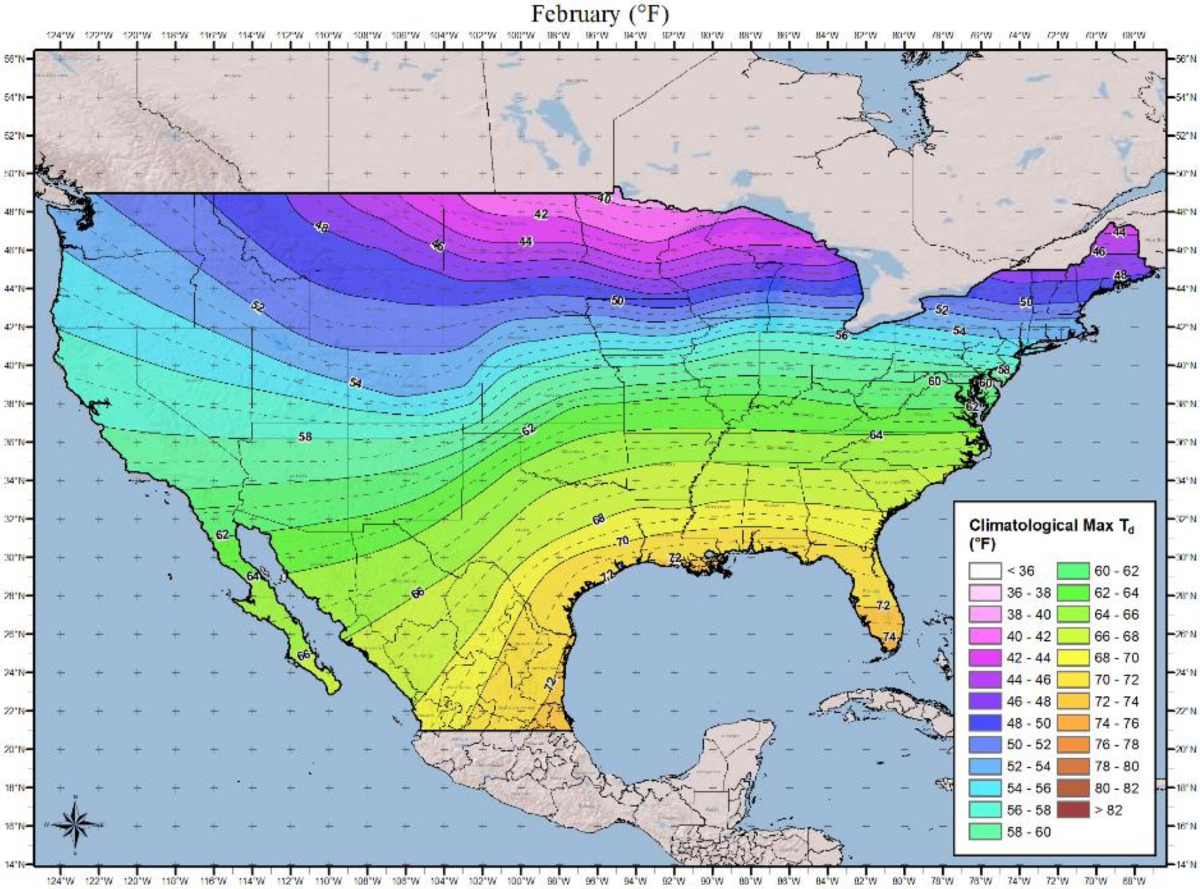


12-hour 1000mb Dew Point Maps

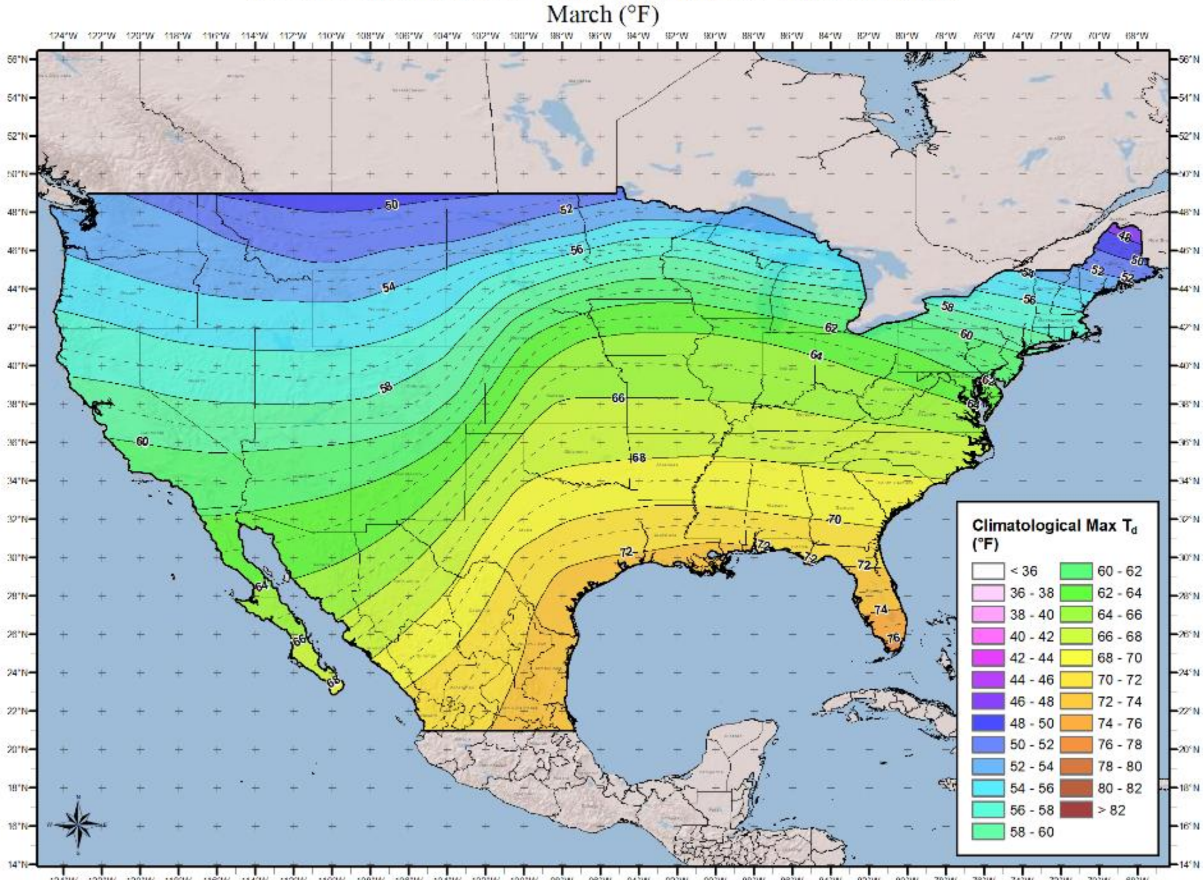
100-year Return Frequency 12-hour Maximum Dew Point Climatology
January (°F)



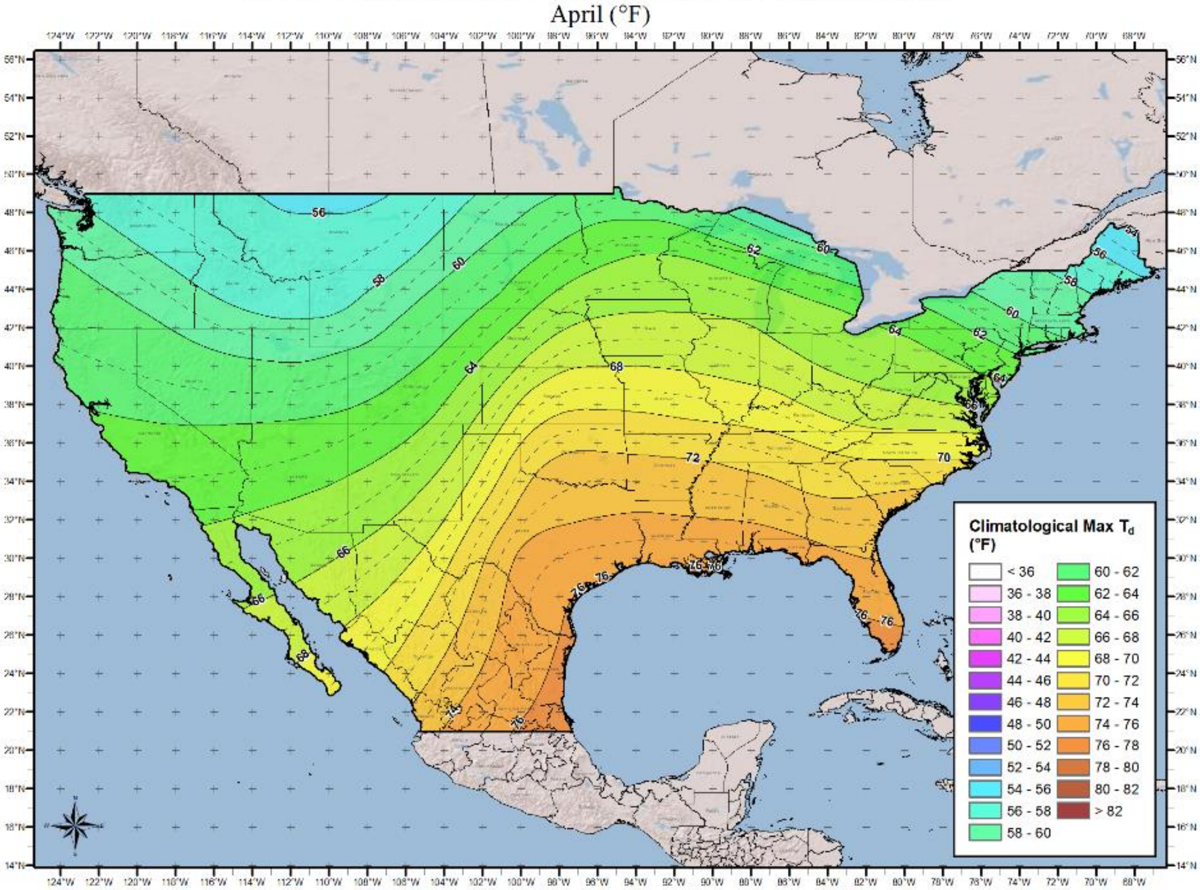
100-year Return Frequency 12-hour Maximum Dew Point Climatology
February (°F)



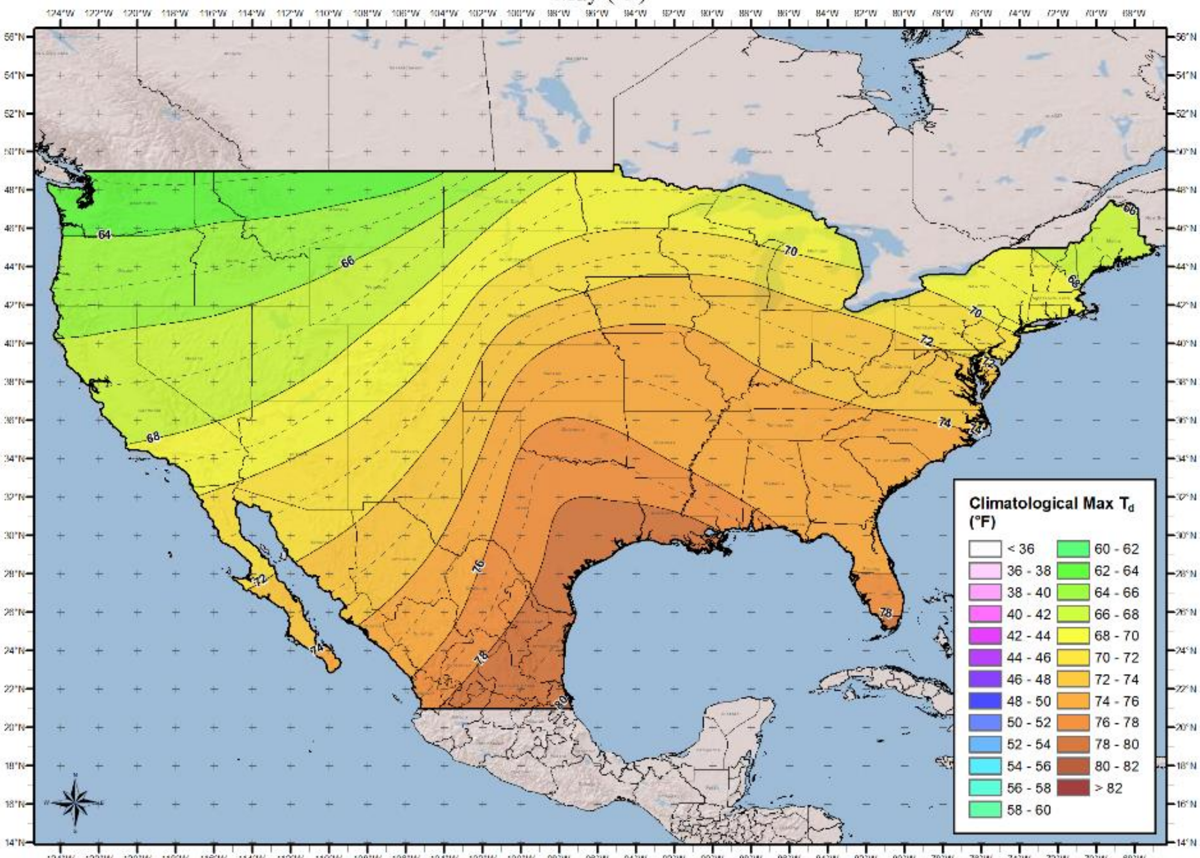
100-year Return Frequency 12-hour Maximum Dew Point Climatology
March (°F)



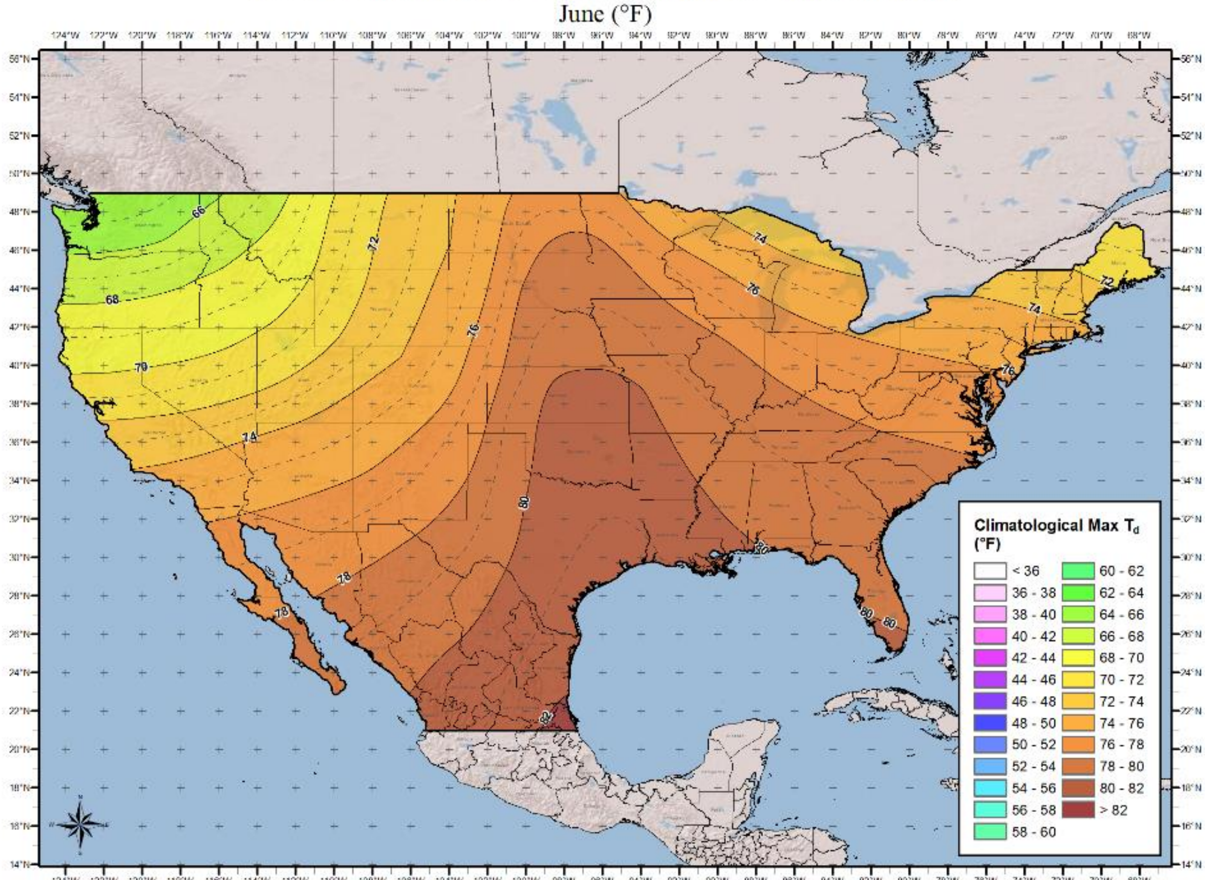
100-year Return Frequency 12-hour Maximum Dew Point Climatology
April (°F)



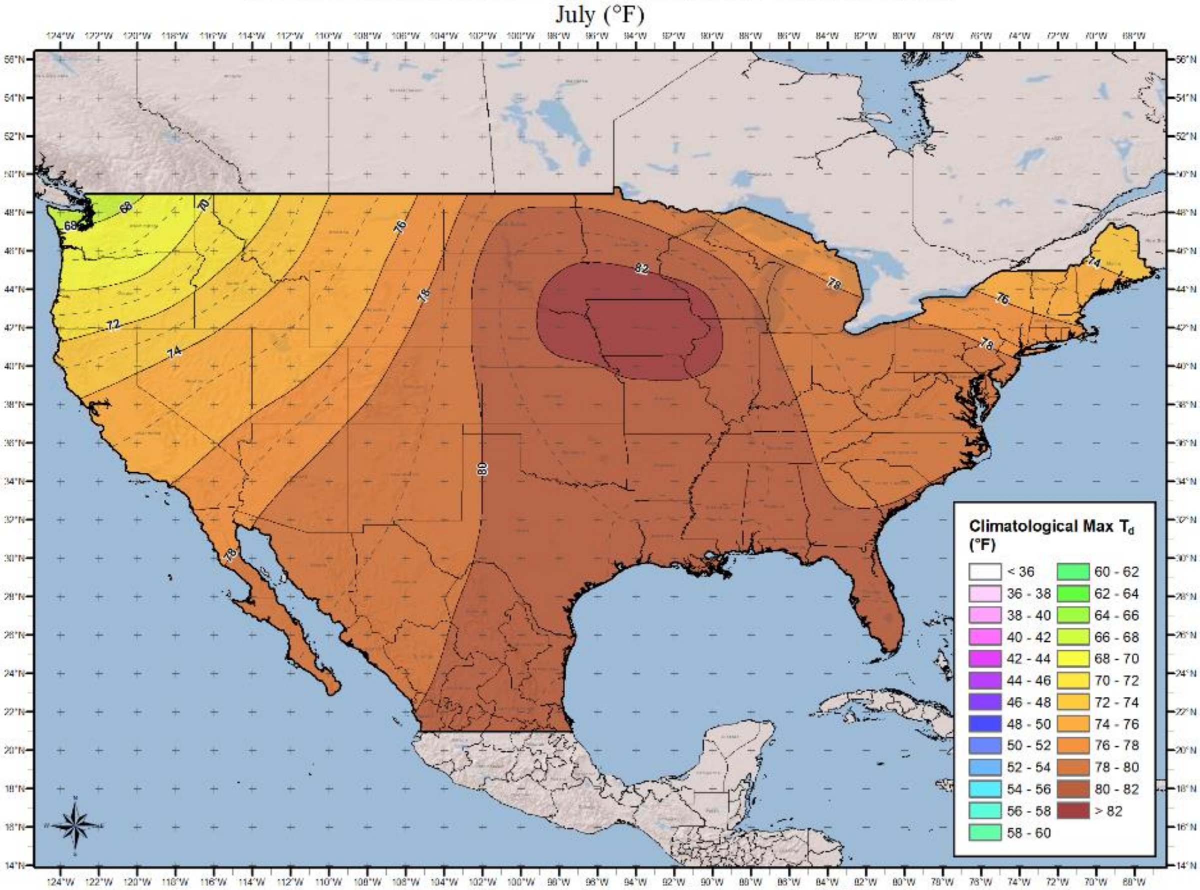
100-year Return Frequency 12-hour Maximum Dew Point Climatology
May (°F)



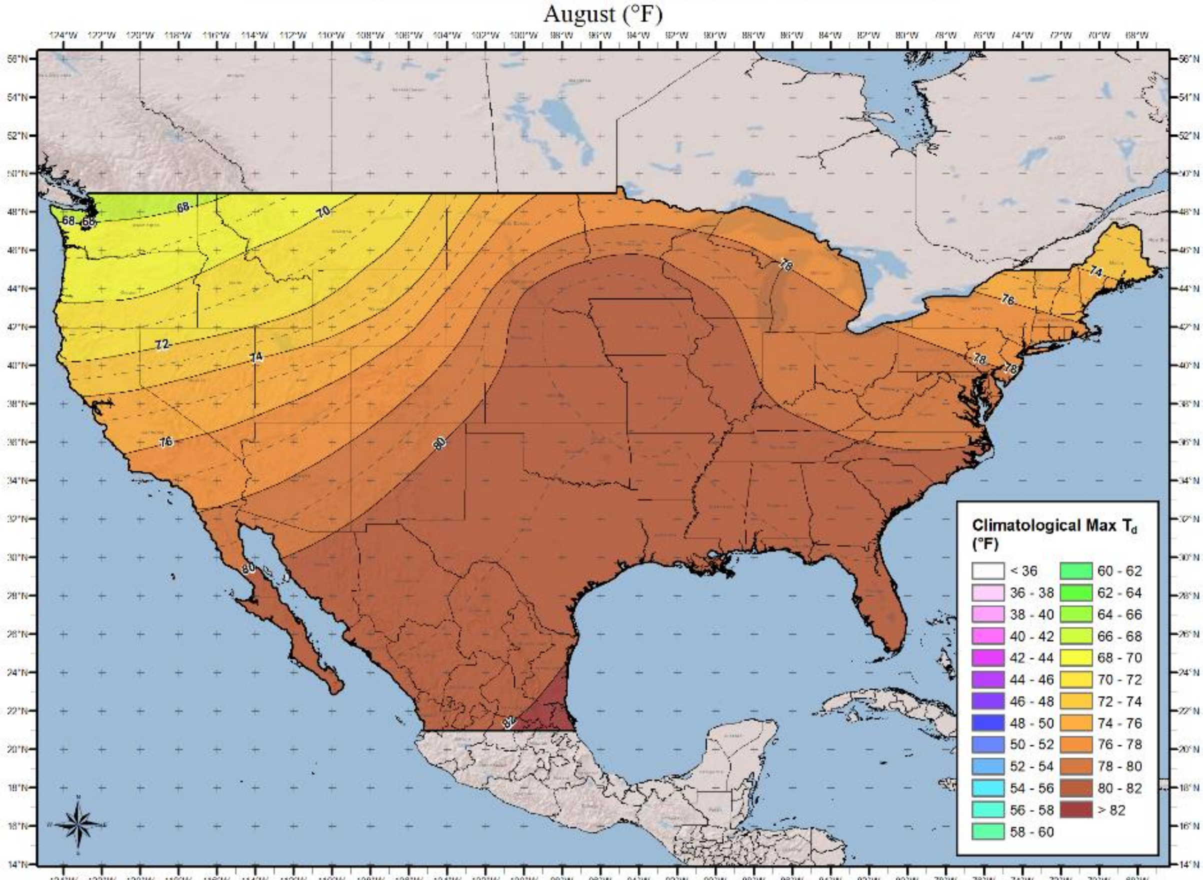
100-year Return Frequency 12-hour Maximum Dew Point Climatology
June (°F)



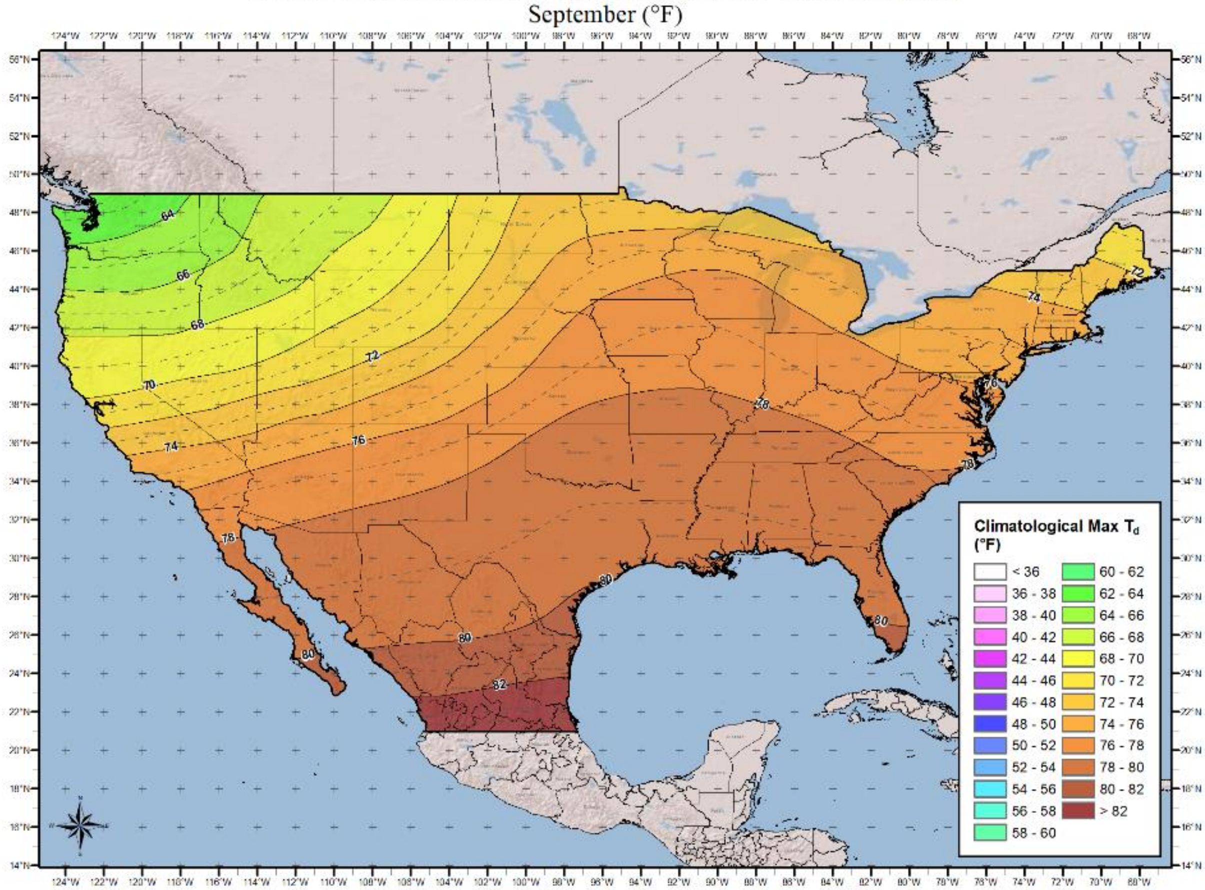
100-year Return Frequency 12-hour Maximum Dew Point Climatology
July (°F)



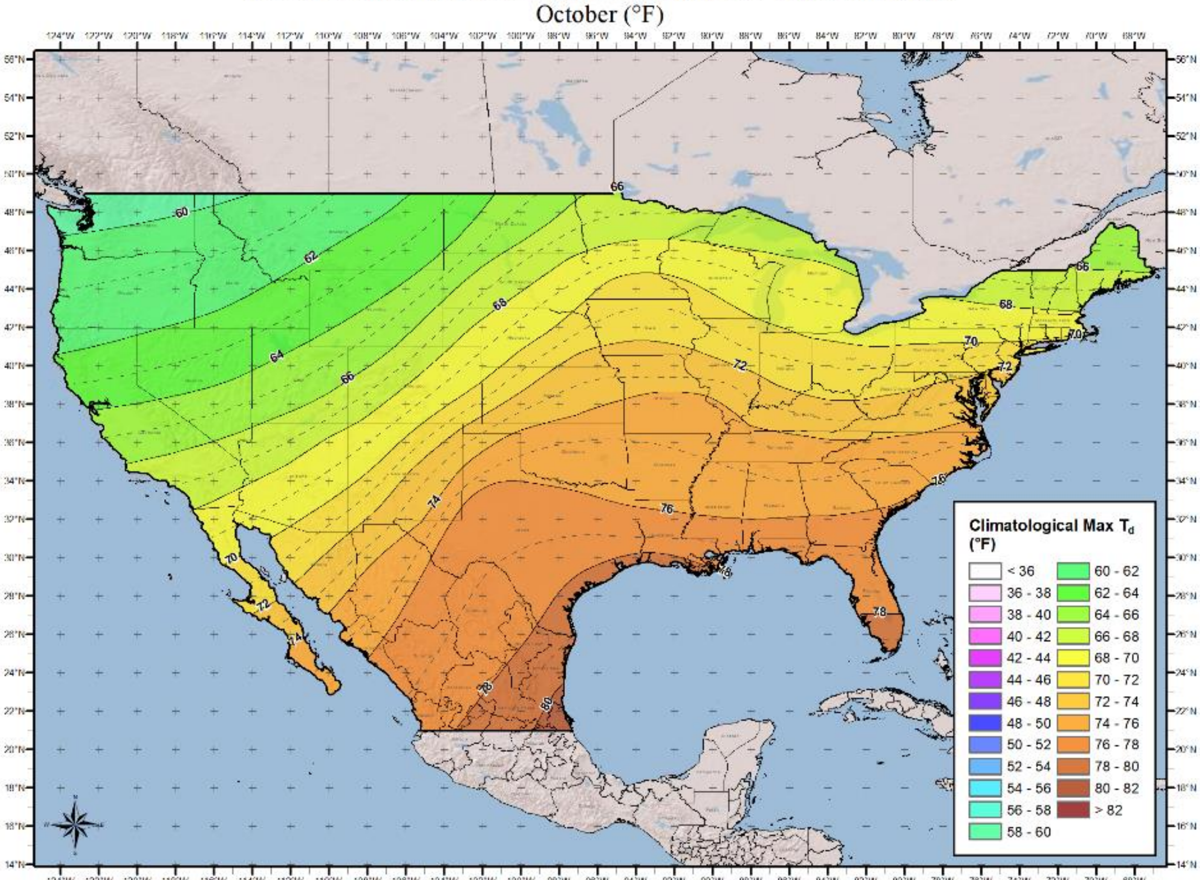
100-year Return Frequency 12-hour Maximum Dew Point Climatology
August (°F)



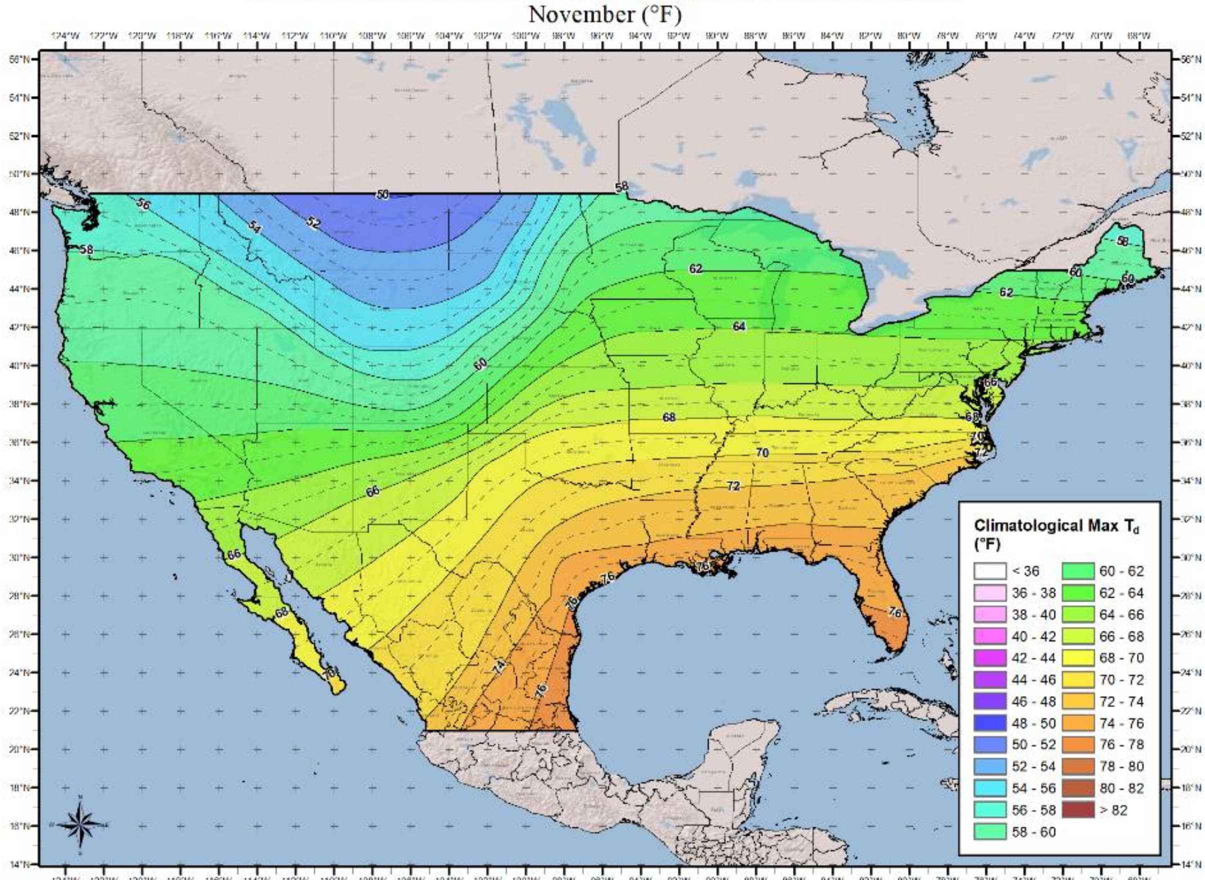
100-year Return Frequency 12-hour Maximum Dew Point Climatology
September (°F)



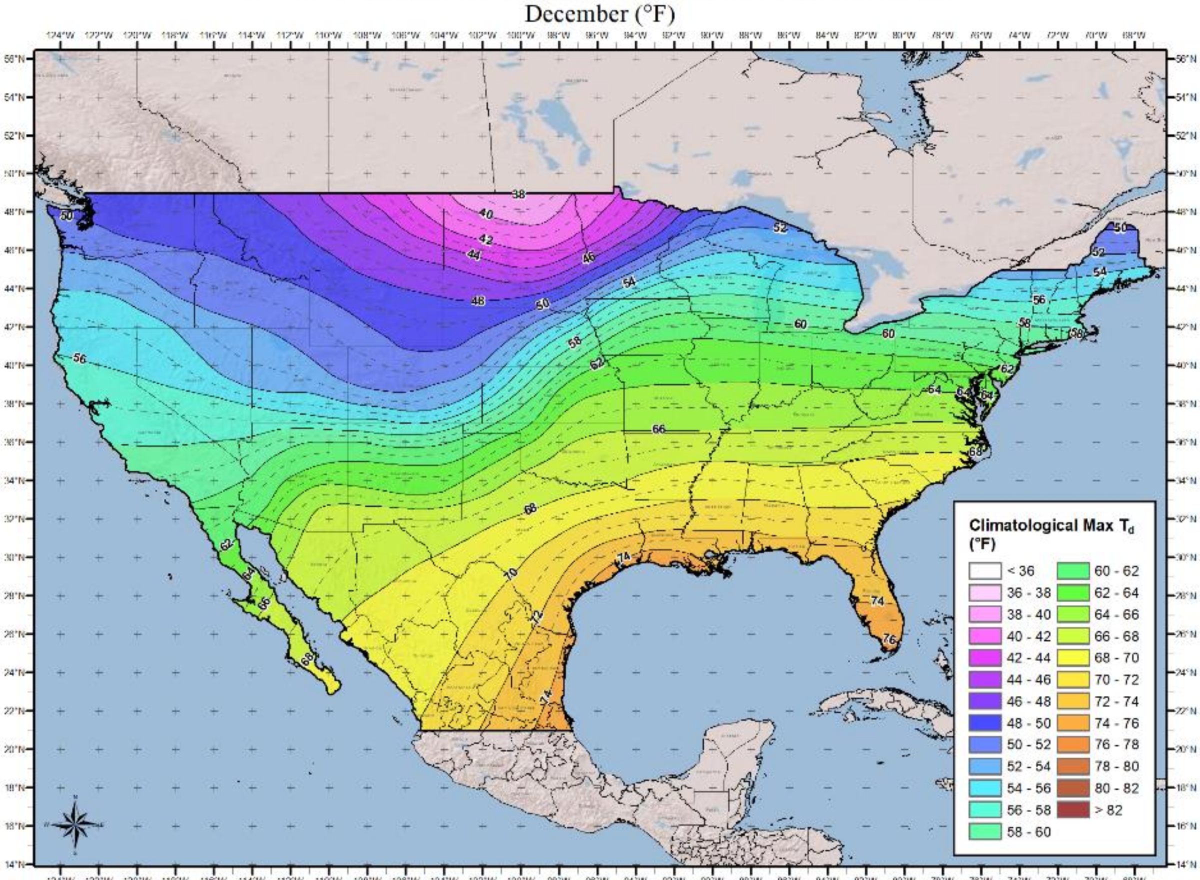
100-year Return Frequency 12-hour Maximum Dew Point Climatology
October (°F)



100-year Return Frequency 12-hour Maximum Dew Point Climatology
November (°F)

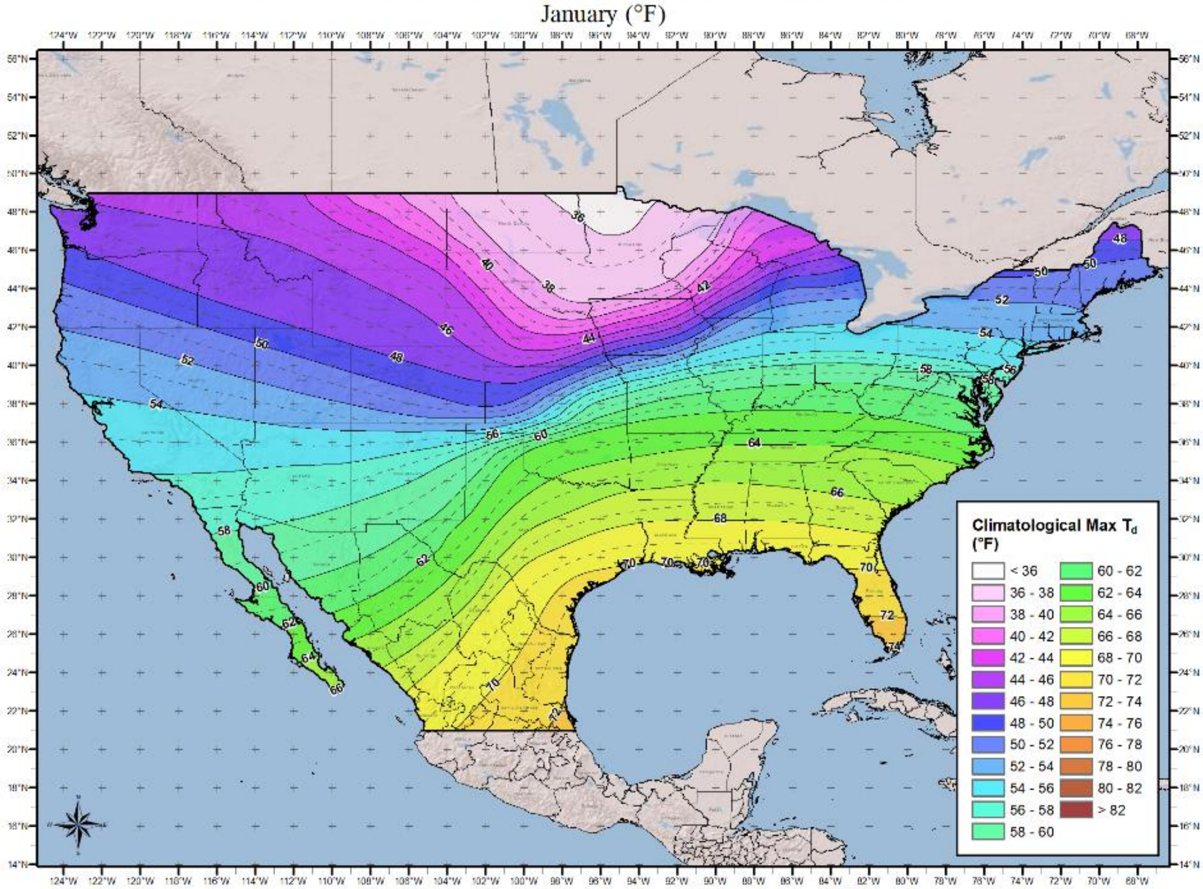


100-year Return Frequency 12-hour Maximum Dew Point Climatology
December (°F)

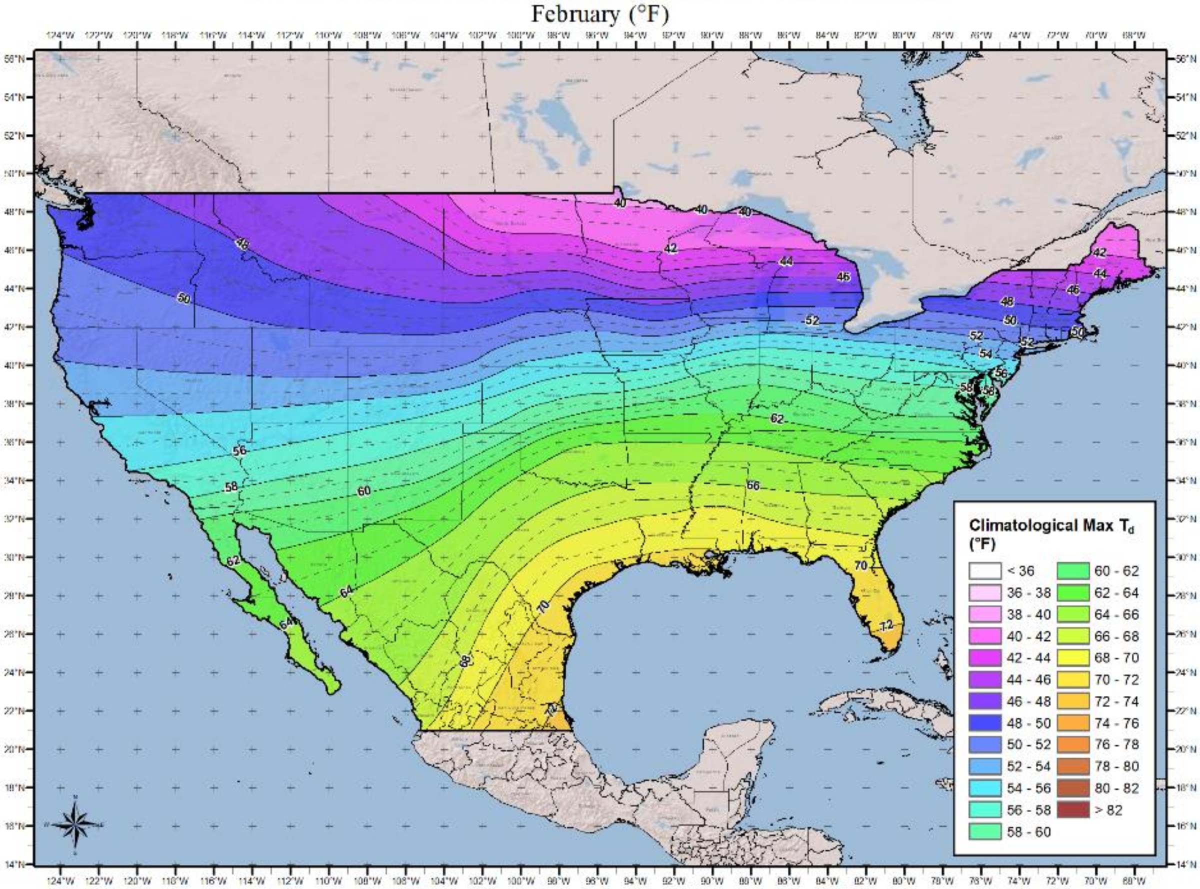


24-hour 1000mb Dew Point Maps

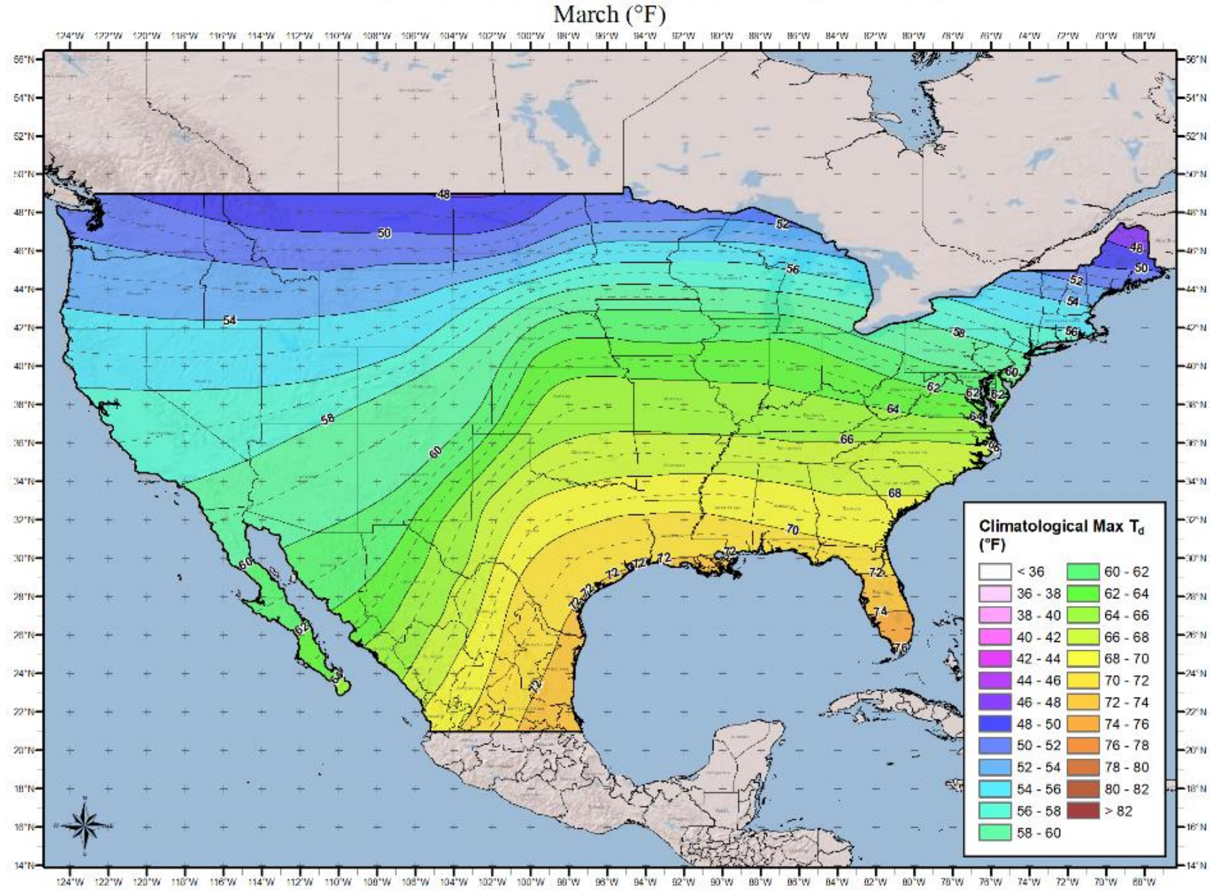
100-year Return Frequency 24-hour Maximum Dew Point Climatology
January (°F)



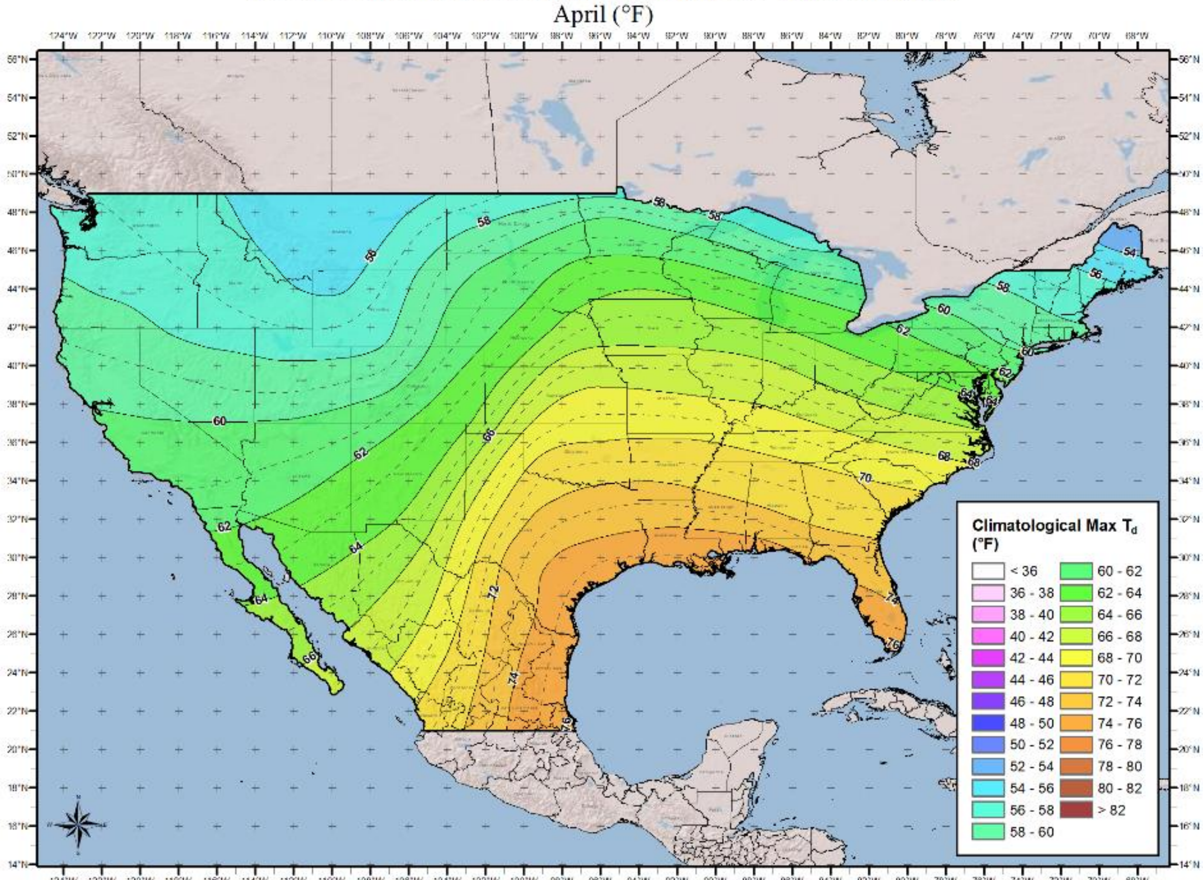
100-year Return Frequency 24-hour Maximum Dew Point Climatology
February (°F)



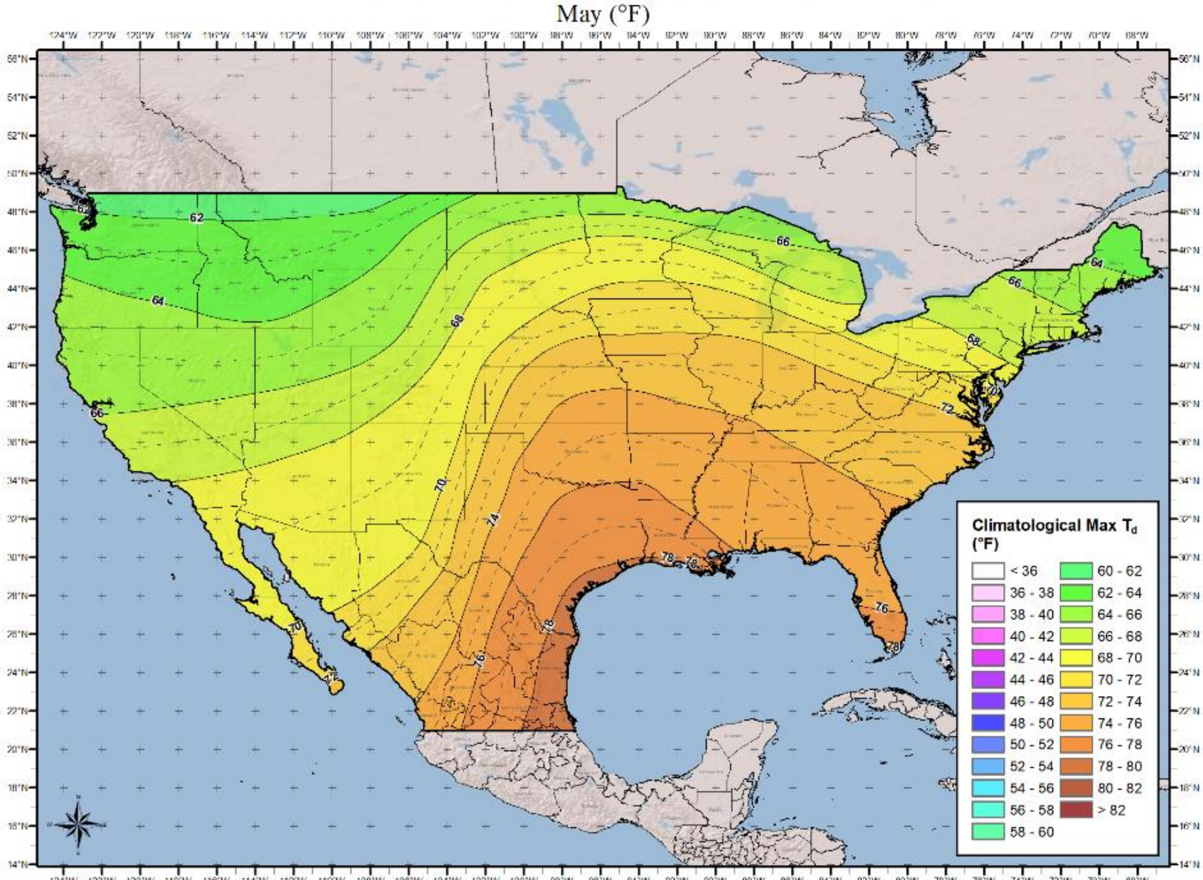
100-year Return Frequency 24-hour Maximum Dew Point Climatology
March (°F)



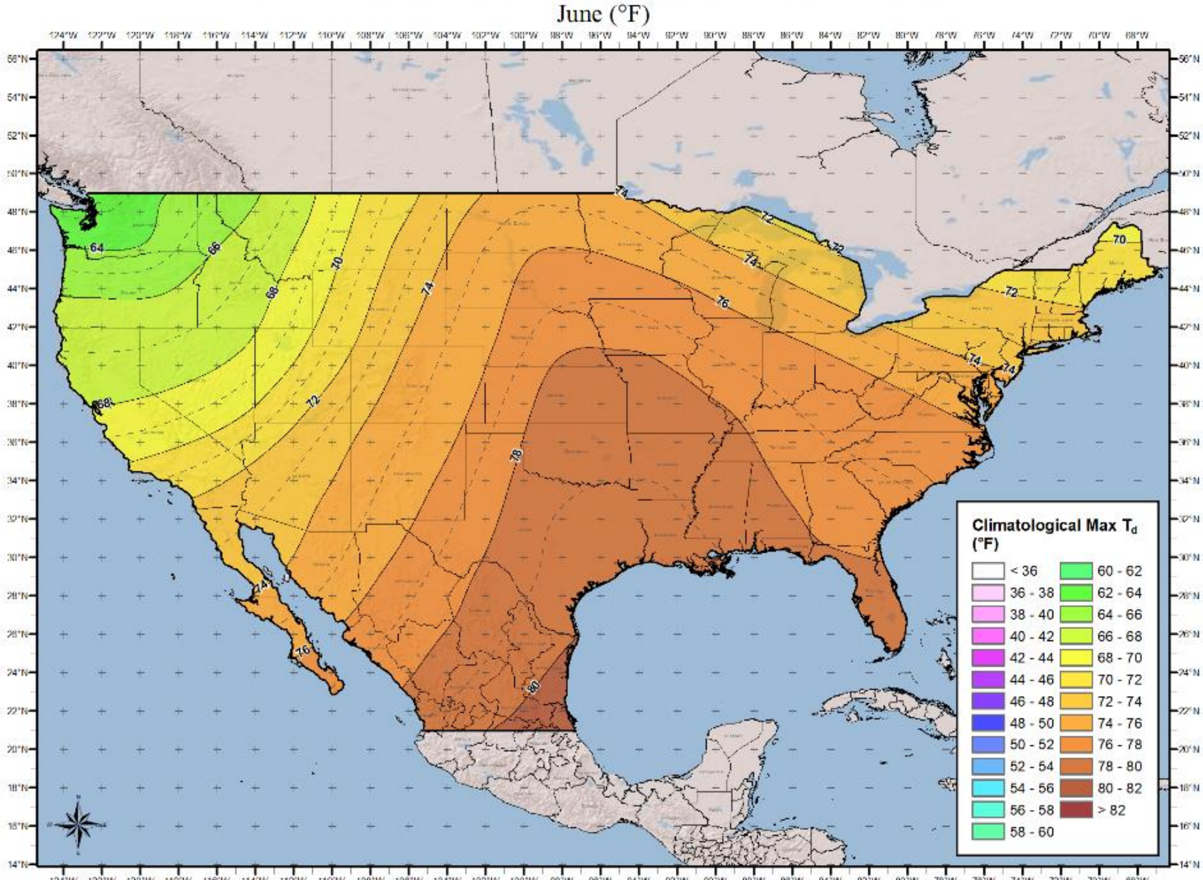
100-year Return Frequency 24-hour Maximum Dew Point Climatology
April (°F)



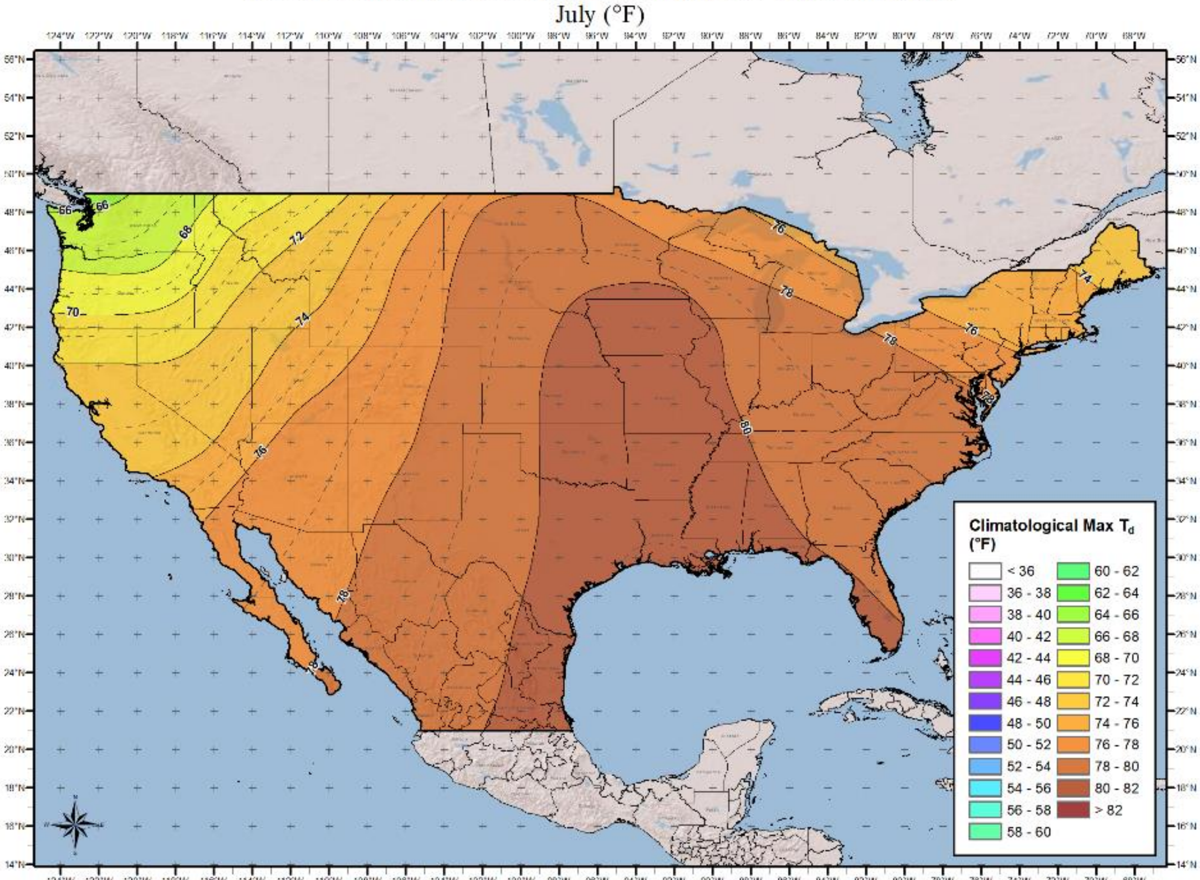
100-year Return Frequency 24-hour Maximum Dew Point Climatology
May (°F)



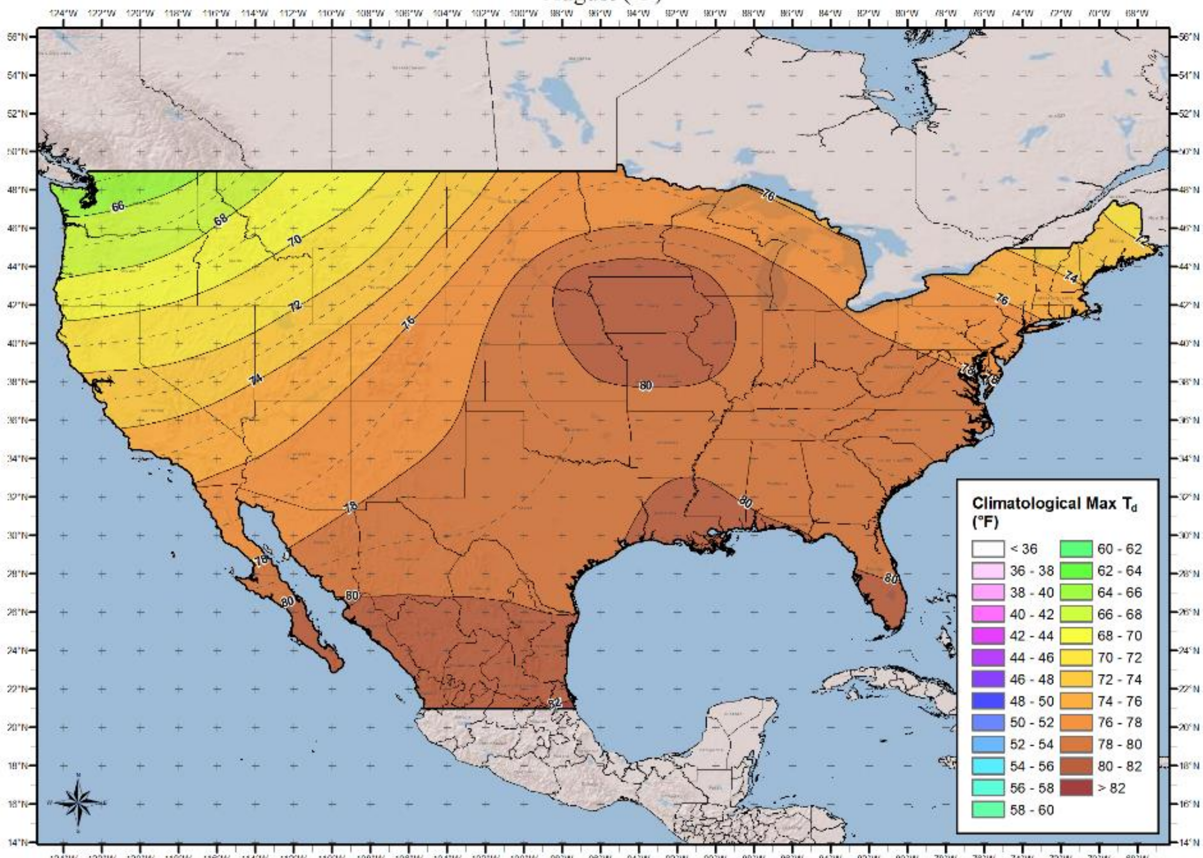
100-year Return Frequency 24-hour Maximum Dew Point Climatology
June (°F)



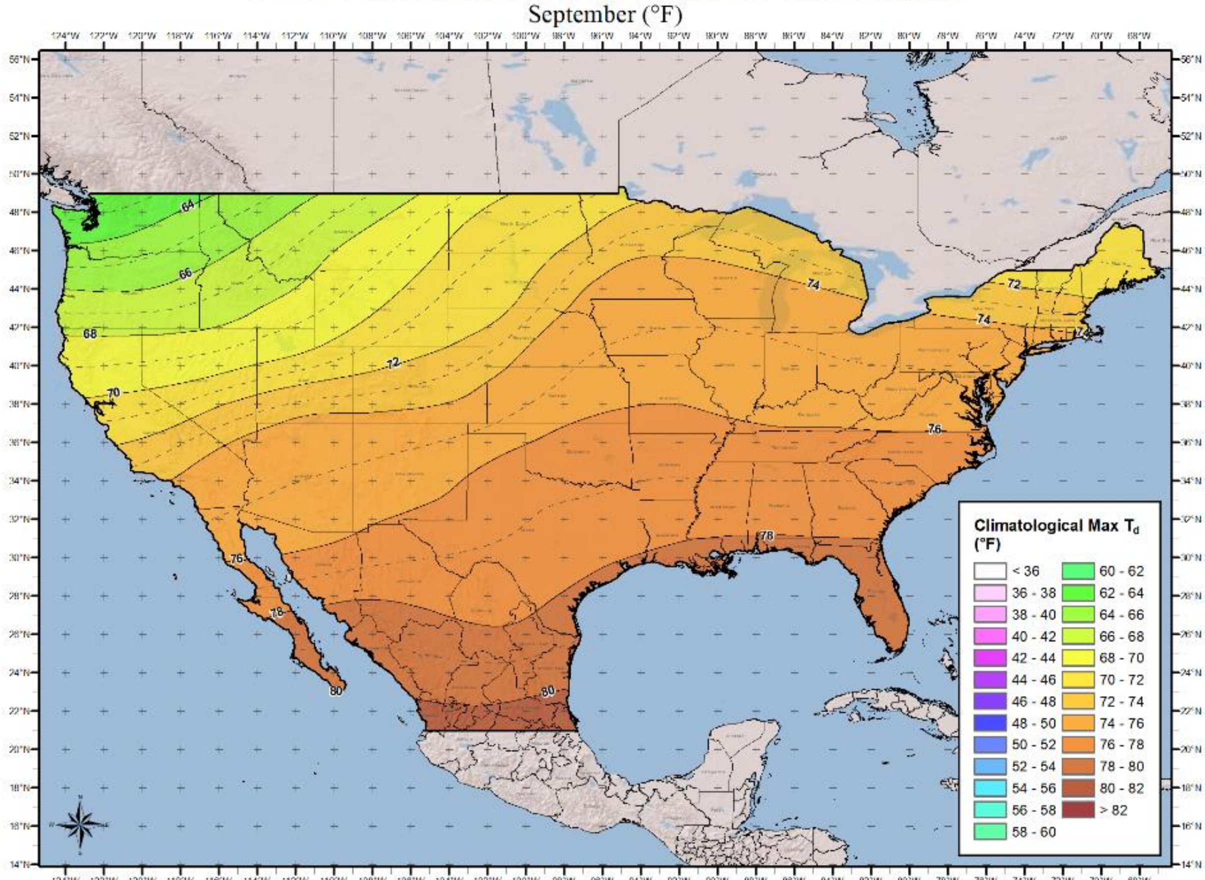
100-year Return Frequency 24-hour Maximum Dew Point Climatology
July (°F)



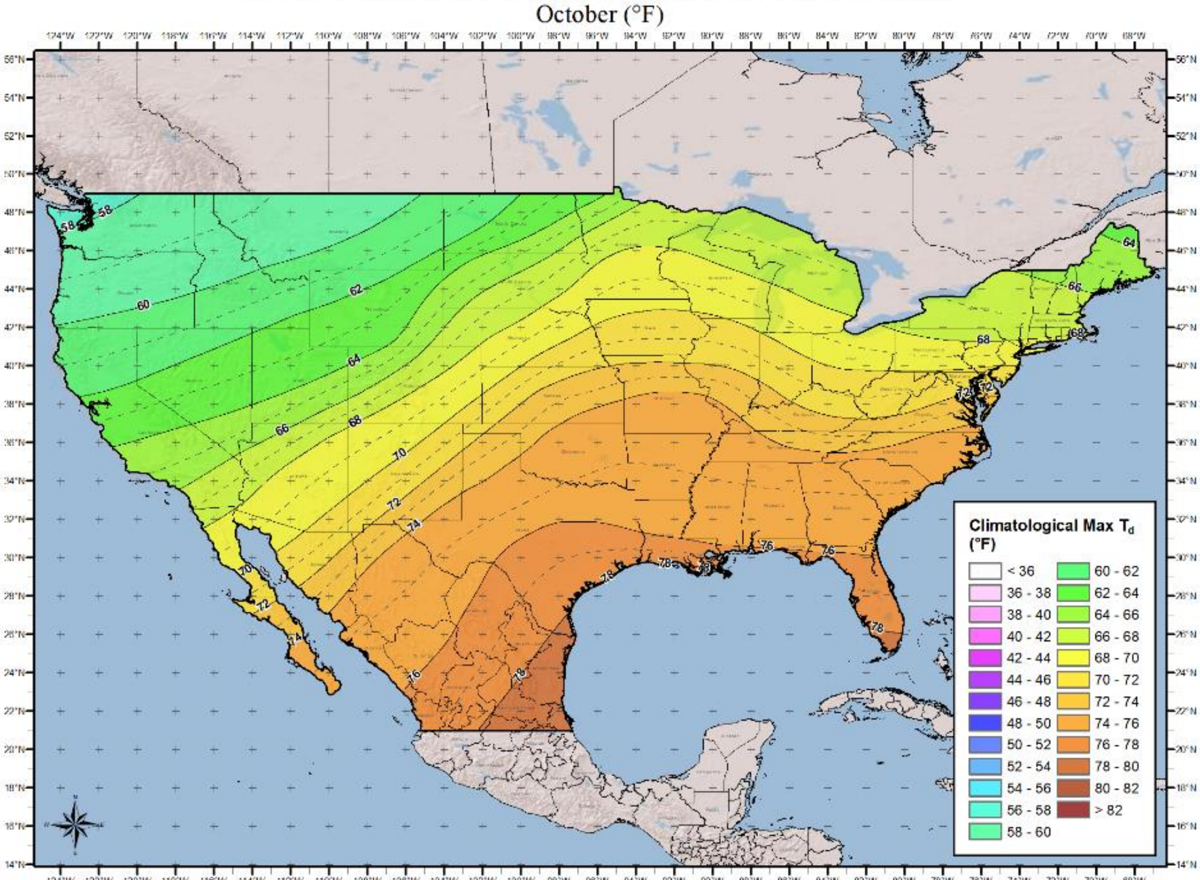
100-year Return Frequency 24-hour Maximum Dew Point Climatology
August (°F)



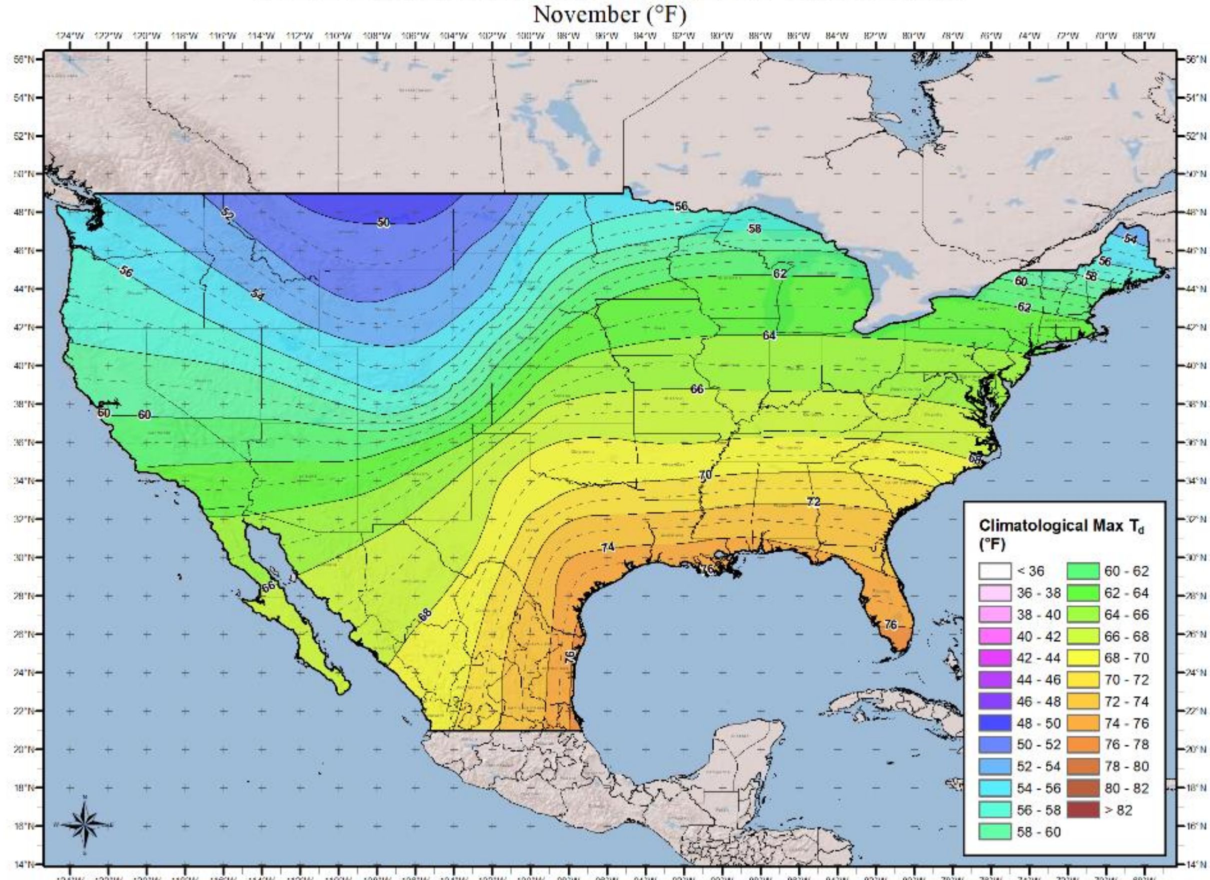
100-year Return Frequency 24-hour Maximum Dew Point Climatology
September (°F)



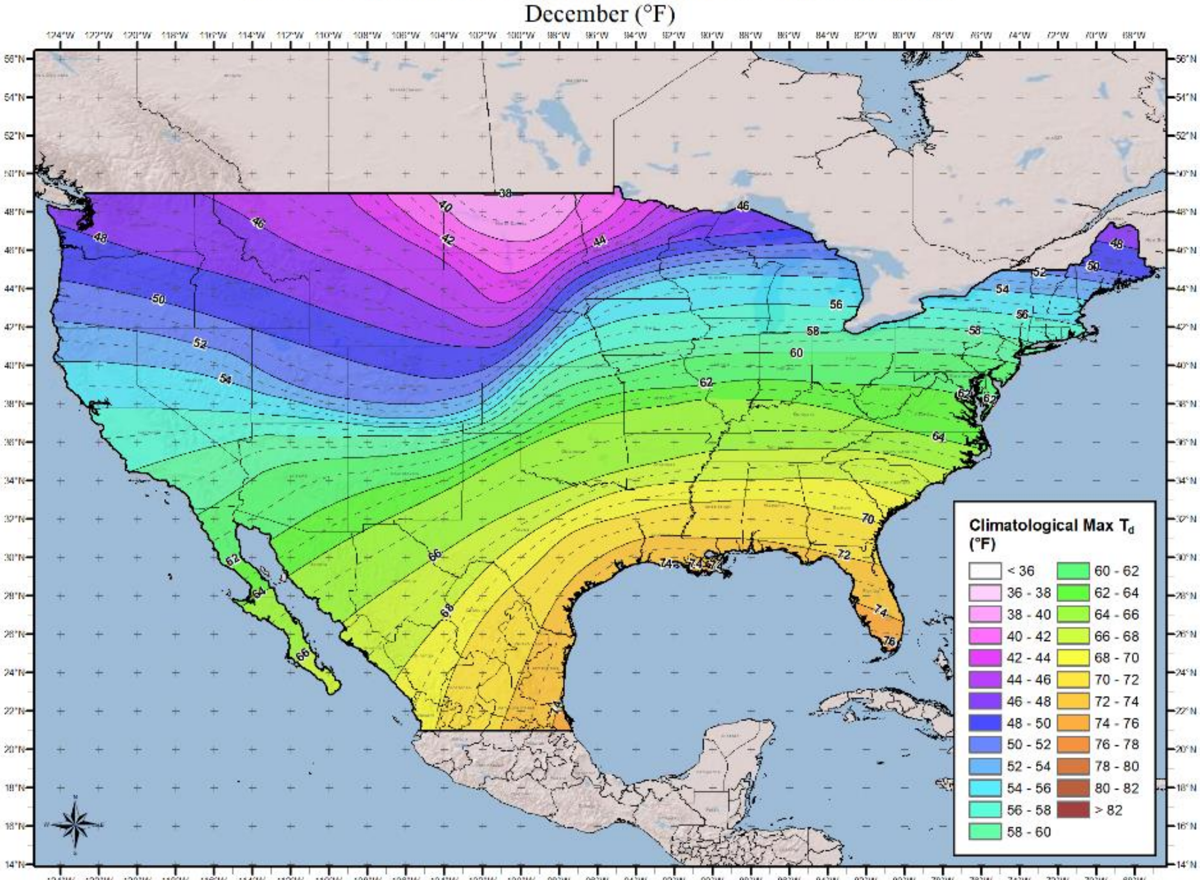
100-year Return Frequency 24-hour Maximum Dew Point Climatology
October (°F)



100-year Return Frequency 24-hour Maximum Dew Point Climatology
November (°F)



100-year Return Frequency 24-hour Maximum Dew Point Climatology
December (°F)



Appendix D
Precipitation Frequency Update Additional Data

The overall PMP domain covers the entire state of North Dakota as well as portions of four adjacent states and two Canadian provinces. Figure 1.1 shows the PMP analysis domain for the state of North Dakota. When calculating the adjustment factors for the PMP analysis, 6- and 24-hour precipitation frequency estimates are used to calculate the GTF. To complete these calculations a consistent precipitation frequency climatology is needed. NOAA Atlas 14 precipitation frequency depths are not available for Montana, Wyoming, Saskatchewan, and Manitoba. Therefore, an updated precipitation frequency climatology was required for these locations that could be combined with the NOAA Atlas 14 data. A new set of 6- and 24-hour precipitation frequency datasets were created during this study and merged with the existing NOAA Atlas 14 datasets to create a seamless precipitation frequency dataset for the entire analysis domain. Figure 1.2 shows the NOAA Atlas 14 depths over the North Dakota PMP domain.

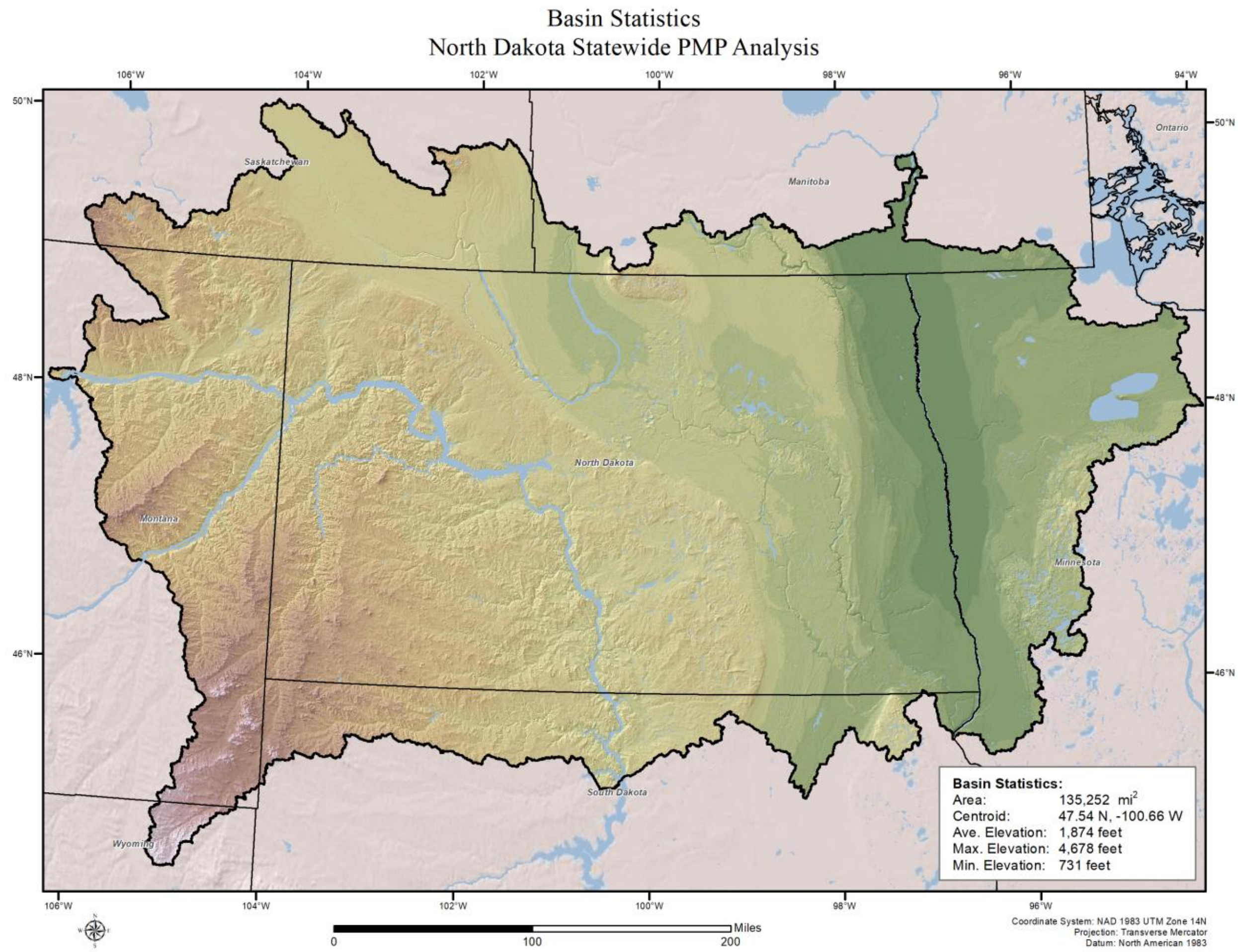


Figure 1.1: PMP analysis domain for North Dakota

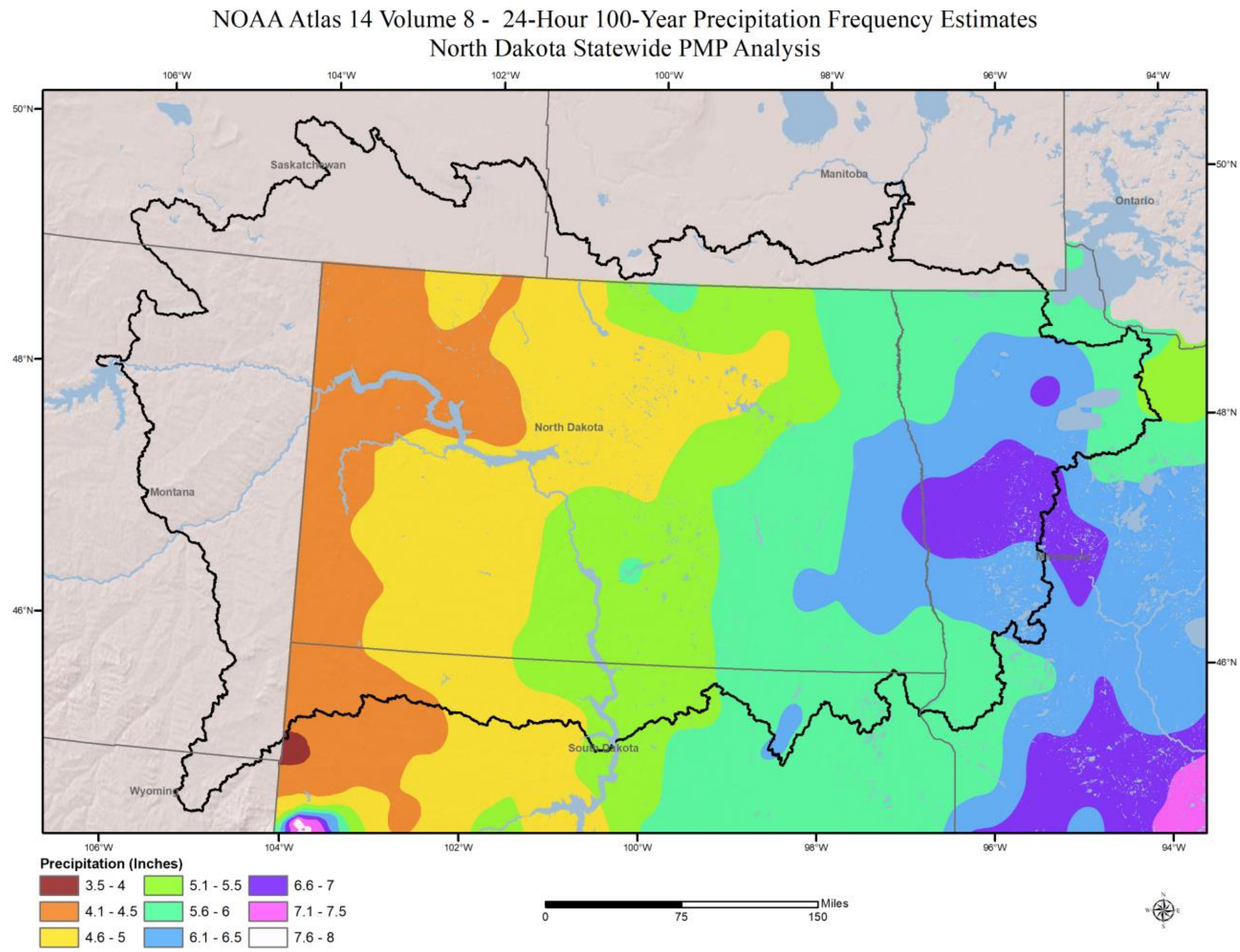


Figure 1.2: 24-Hour 100-Year NOAA Atlas 14 precipitation frequency estimates over the PMP domain

Regional Frequency Analyses Methods

Regional precipitation frequency analysis was conducted for the North Dakota domain to provide precipitation frequency estimates for application in GTF, PMP, and hydrologic modeling. Precipitation frequency estimates were created for two durations (6-hour, and 24-hour) and ten frequencies (1-, 2-, 5-, 10-, 25-, 50-, 100-, 200-, 500-, and 1000-year). Hourly (94 stations) and daily (286 stations) station data were extracted for the two durations from Environment Canada and the National Weather Service (NWS). This initial regional analysis used two climatic regions and tested for homogeneity, i.e. if regions are homogenous they statistically represent similar meteorology and can be modeled based on the same regional probability distribution "Regional Growth Curve". Hosking and Wallis (1997) developed heterogeneity measures to help indicate the level of heterogeneity or homogeneity in the L-moment ratios for a group of stations representing a sub-region. The statistics H1 (heterogeneity measure) and H2 denote the relative variability of observed L-Cv and L-Skewness respectively for stations within a sub-region. As suggested in Hosking and Wallis (1997), adjustments of regions, such as moving stations from one region to another or subdividing a region, were made to reduce heterogeneity.

The heterogeneity measure, H1, tests between-site variations in sample L-moments for a group of sites with what would be expected for a homogeneous region based on coefficient of L-variation (Hosking and Wallis, 1997). Earlier studies (Hosking and Wallis, 1997 and Bonnin et al., 2004) indicated that a threshold of 2 is conservative and Schaefer et al., (2006) indicated that a threshold of 3.0 is conservative. For the North Dakota project a H1 threshold of 2.0 was used to identify homogeneity. The initial climatic two regions contained enough data to perform reliable homogeneity tests, with the H1 statistics for the two durations having homogeneity (H < 2).

L-moments statistics using R-statistical software packages lmom and lmomRFA developed by Hosking (Hosking, 2015a, and Hosking 2015b) were used. L-moment statistics are used for computing sample statistics for data at individual sites; for testing for homogeneity/heterogeneity of proposed groupings of sites (regions); for conducting goodness-of-fit tests for identifying a suitable probability distribution(s); and for solving for distribution parameters for the selected probability distribution. L-moments obtain their name from their construction as linear combinations of order statistics (Hosking and Wallis, 1997).

L-moment statistics are a significant improvement over conventional product moment statistics for characterizing the shape of a probability distribution and estimating the distribution parameters, particularly for environmental data where sample sizes are commonly small. Unlike product moments, the sampling properties for L-moments statistics are nearly unbiased, even in small samples, and are near normally distributed. These properties make them well suited for characterizing environmental data that commonly exhibit moderate to high skewness. The L-moment measure of location, and L-moment ratio measures of scale, skewness, and kurtosis are calculated based on Hosking and Wallis (1997). Goodness-of-fit measures were evaluated for five candidate distributions: generalized logistic (GLO), generalized extreme value (GEV), generalized normal (GNO), Pearson type III (PE3), and generalized Pareto (GPA). An L-Moment Ratio Diagram was prepared for each duration based on L-Skewness and L-Kurtosis pairs for the stations used.

The regional weighted-average L-Skewness and L-Kurtosis pairing found the GEV, , and GNO distributions to be the most frequent distributions that were statistically significant based on goodness-of-fit test. The GEV distribution was statistically significant for all durations (1st), whereas the GNO distribution was also significant. Based on the goodness-of-fit statistics and summary data, the GEV distribution was selected for derivation of the precipitation AEPs.

The GEV is a general mathematical form that incorporates Gumbel's Extreme Value (EV) type I, II, and III distributions for maxima. The parameters of the GEV distribution are the ξ (location), α (scale), and k (shape). The Gumbel EV type I distribution is obtained when $k = 0$. For $k > 0$, the distribution has finite upper bound at $\xi + \alpha / k$ and corresponds to the EV type III distribution for maxima that are bounded above. For $k < 0$, this corresponds to the Gumbel EV type II distribution.

Utilizing regional methods described in Hosking and Wallis (1997) together with quality-controlled annual maximum precipitation values extracted for stations within each region, regional L-moment statistics were computed and applied to derive precipitation frequency estimates. Consistent with methodologies used in United States Precipitation Frequency climatology (NOAA Atlas 14, Bonnin et al., 2004; Perica et al., 2013), the station precipitation frequency estimates were spatially interpolated utilizing a climatologically-aided interpolation approach.

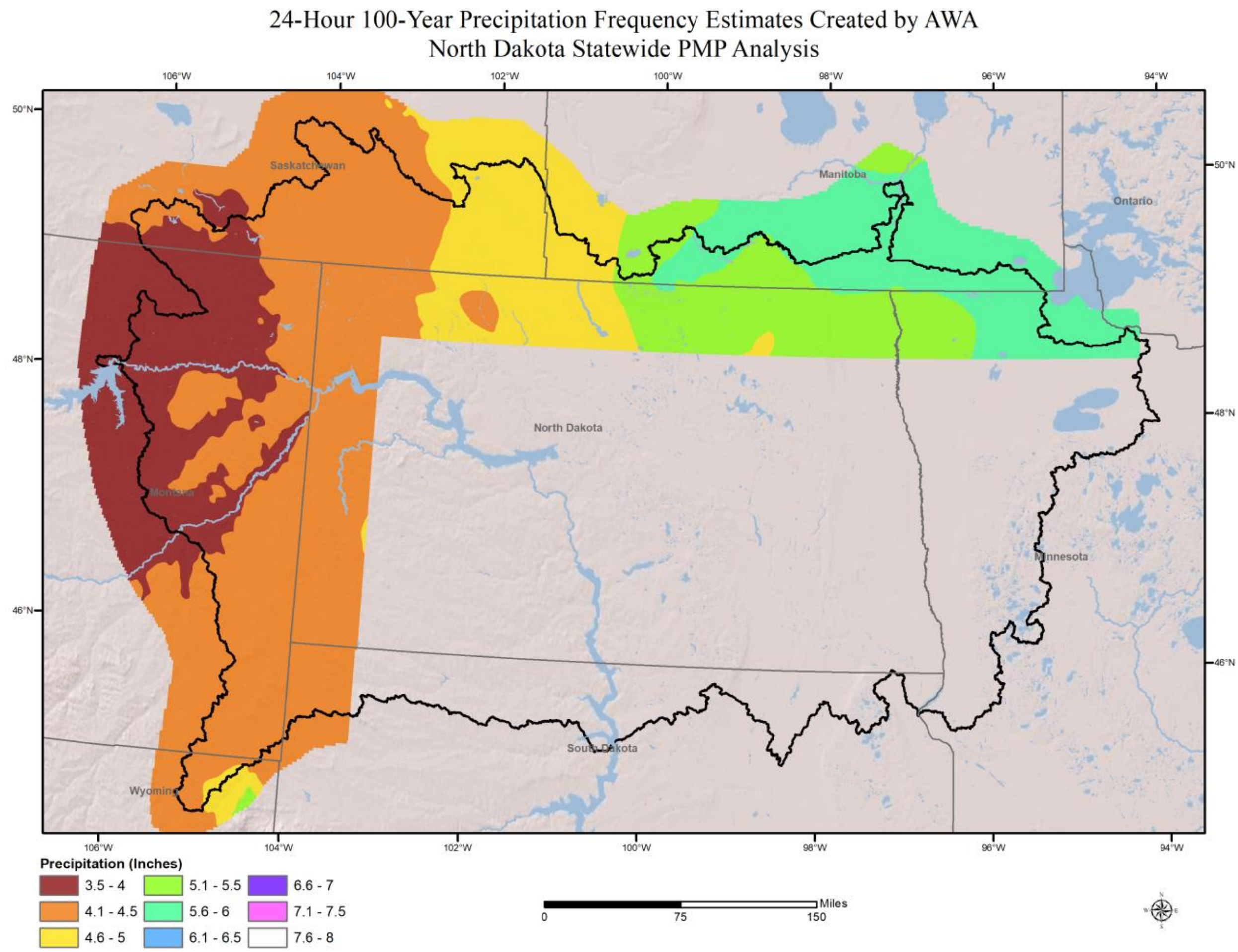


Figure 1.3: 24-Hour 100-Year precipitation frequency estimates created

Since the NOAA Atlas 14 and North Dakota frequency datasets were completed independently from each other small inconsistencies occurred along the border. The goal was to leave the NOAA Atlas 14 depths unchanged and to seamlessly merge the two datasets together. To accomplish this, the newly created precipitation frequency dataset was clipped back 15 miles outside of the boundary of the NOAA Atlas 14 domain. The NOAA 14 depths were left as is. Figure 1.4 shows the clipped area and both datasets ready to merge.

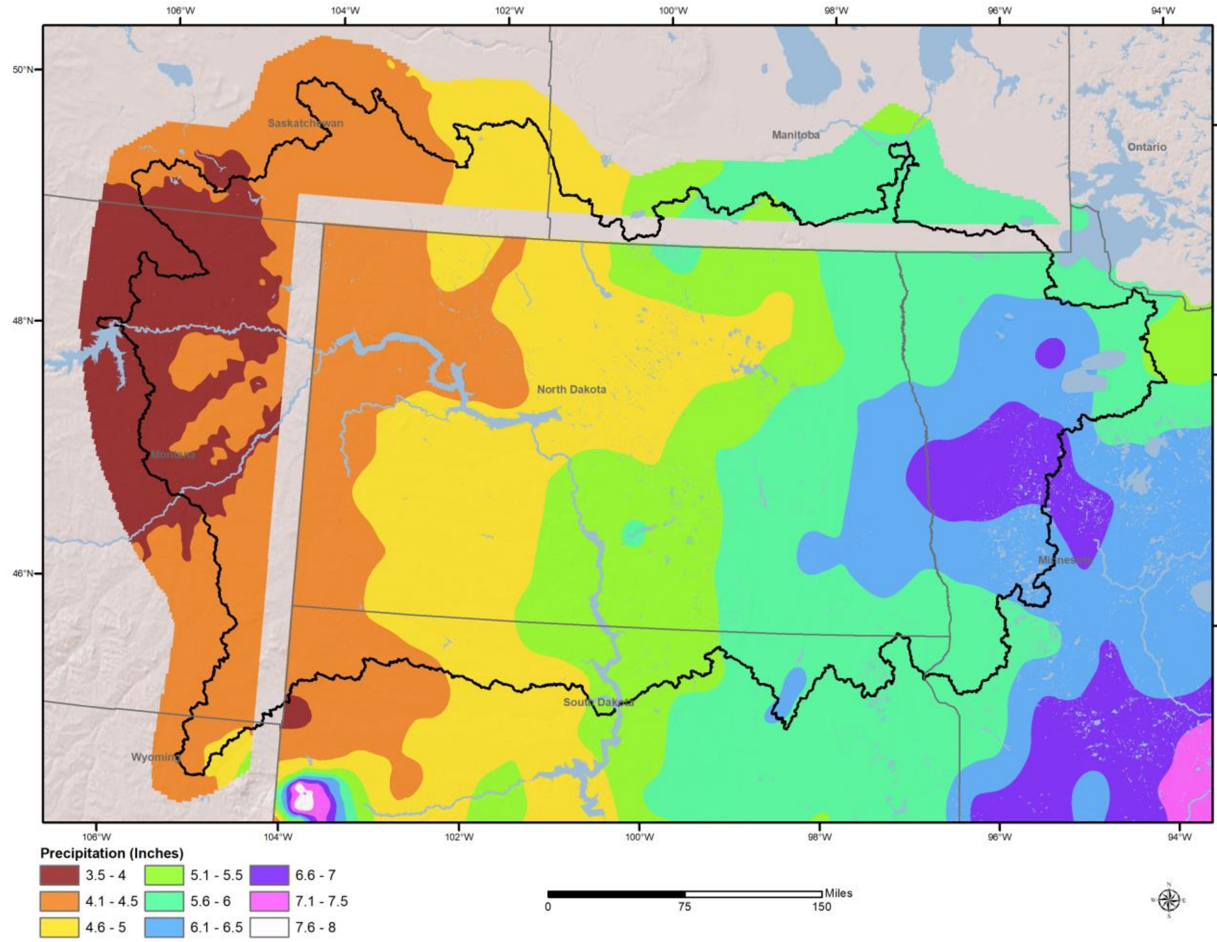


Figure 1.4: Area to be interpolated between datasets

At this point, one tenth of an inch contours were created for both datasets. A GIS interpolation method was used to fill in the fifteen-mile buffer zone between the two datasets. This resulted in a seamless dataset with the buffer area filled in with little to no changes to the existing datasets. Figure 1.5 shows the final 6-hour 100-year precipitation and Figure 1.6 show the final 24-hour 100-year precipitation frequency estimates used for the GTF calculations.

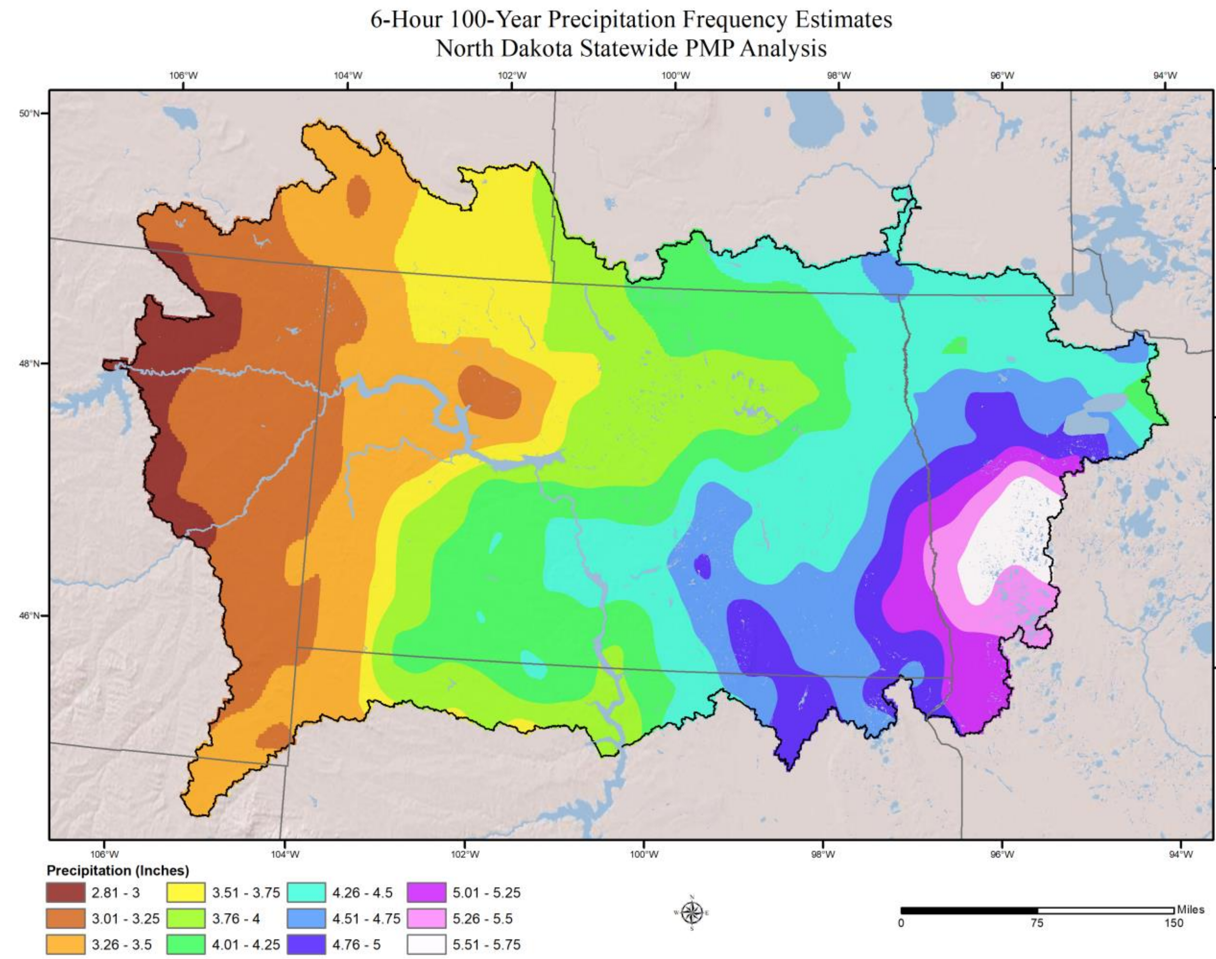


Figure 1.5: Final 6-Hour 100-Year precipitation frequency estimates over the North Dakota PMP domain

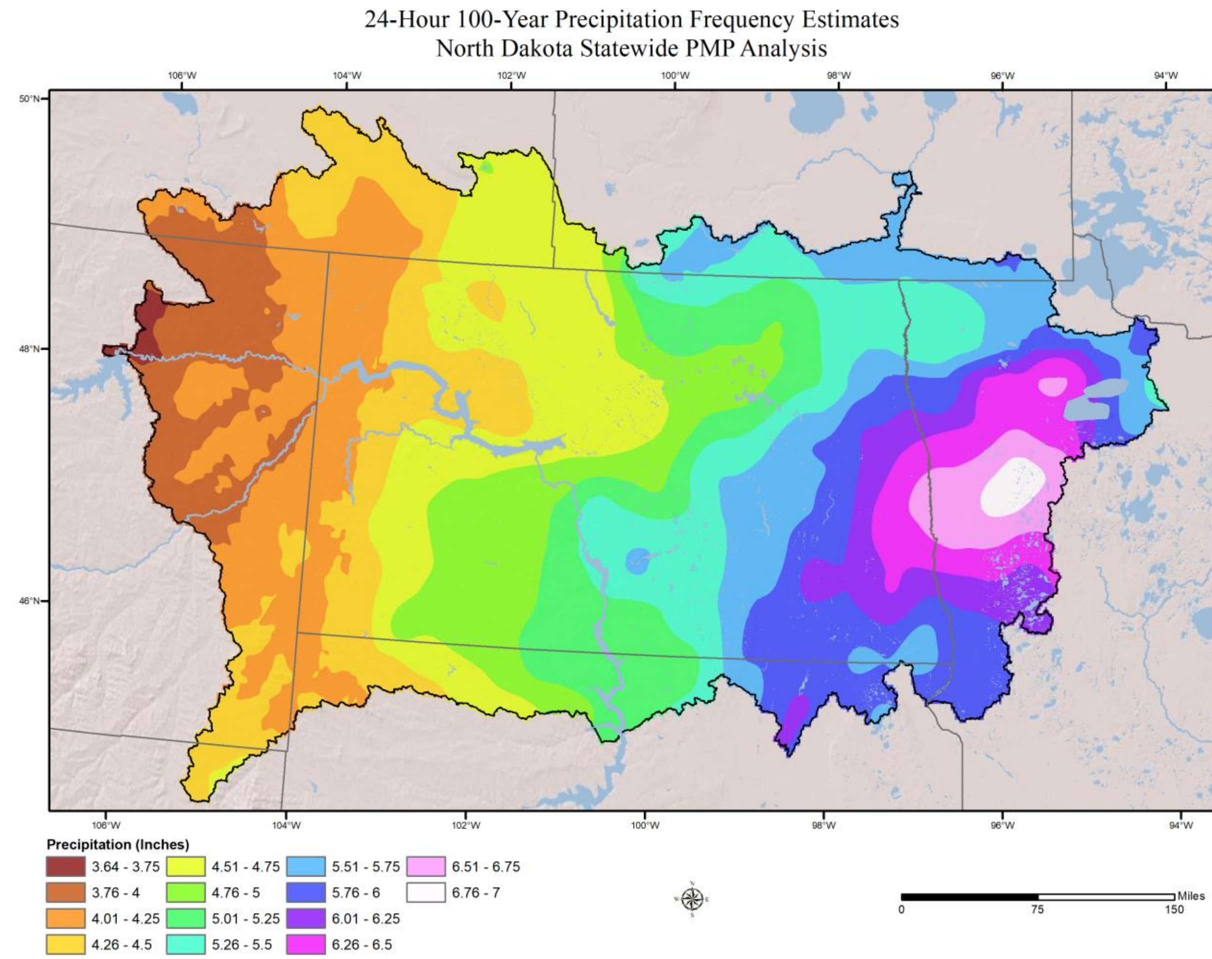


Figure 1.6: Final 24-Hour 100-Year precipitation frequency estimates over the North Dakota PMP domain

Appendix E
Storm Precipitation Analysis System (SPAS) Description

Introduction

The Storm Precipitation Analysis System (SPAS) is grounded on years of scientific research with a demonstrated reliability in hundreds of post-storm precipitation analyses. It has evolved into a trusted hydrometeorological tool that provides accurate precipitation data at a high spatial and temporal resolution for use in a variety of sensitive hydrologic applications (Faulkner et al., 2004, Tomlinson et al., 2003-2012). Applied Weather Associates, LLC and METSTAT, Inc. initially developed SPAS in 2002 for use in producing Depth-Area-Duration values for Probable Maximum Precipitation (PMP) analyses. SPAS utilizes precipitation gauge data, basemaps and radar data (when available) to produce gridded precipitation at time intervals as short as 5 minutes, at spatial scales as fine as 1 km^2 and in a variety of customizable formats. To date (March 2015 SPAS has been used to analyze over 500 storm centers across all types of terrain, among highly varied meteorological settings and some occurring over 100-years ago.

SPAS output has many applications including, but not limited to: hydrologic model calibration/validation, flood event reconstruction, storm water runoff analysis, forensic cases and PMP studies. Detailed SPAS-computed precipitation data allow hydrologists to accurately model runoff from basins, particularly when the precipitation is unevenly distributed over the drainage basin or when rain gauge data are limited or not available. The increased spatial and temporal accuracy of precipitation estimates has eliminated the need for commonly made assumptions about precipitation characteristics (such as uniform precipitation over a watershed), thereby greatly improving the precision and reliability of hydrologic analyses.

To instill consistency in SPAS analyses, many of the core methods have remained consistent from the beginning. However, SPAS is constantly evolving and improving through new scientific advancements and as new data and improvements are incorporated. This write-up describes the current inner-workings of SPAS, but the reader should realize SPAS can be customized on a case-by-case basis to account for special circumstances; these adaptations are documented and included in the deliverables. The over-arching goal of SPAS is to combine the strengths of rain gauge data and radar data (when available) to provide sound, reliable and accurate spatial precipitation data.

Hourly precipitation observations are generally limited to a small number of locations, with many basins lacking observational precipitation data entirely. However, Next Generation Radar (NEXRAD) data provide valuable spatial and temporal information over data-sparse basins, which have historically lacked reliability for determining precipitation rates and reliable quantitative precipitation estimates (QPE). The improved reliability in SPAS is made possible by hourly calibration of the NEXRAD radar-precipitation relationship, combined with local hourly bias adjustments to force consistency between the final result and “ground truth” precipitation measurements. If NEXRAD radar data are available (generally for storm events since the mid-1990s), precipitation accumulation at temporal scales as frequent as 5-minutes can be analyzed. If no NEXRAD data are available, then precipitation data are analyzed in hourly increments. A summary of the general SPAS processes is shown in flow chart in Figure E.1.

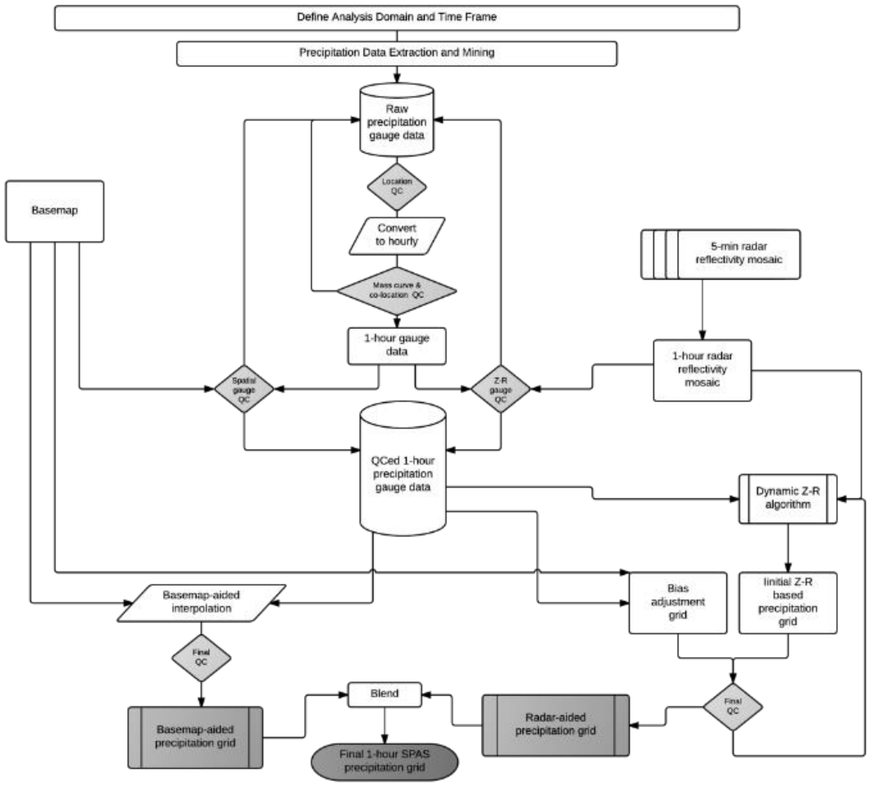


Figure E.1: SPAS flow chart

Setup

Prior to a SPAS analysis, careful definition of the storm analysis domain and time frame to be analyzed is established. Several considerations are made to ensure the domain (longitude-latitude box) and time frame are sufficient for the given application.

SPAS Analysis Domain

For PMP applications it is important to establish an analysis domain that completely encompasses a storm center, meanwhile hydrologic modeling applications are more concerned about a specific basin, watershed or catchment. If radar data are available, then it is also important to establish an area large enough to encompass enough stations (minimum of ~30) to adequately derive reliable radar-precipitation intensity relationships (discussed later). The domain is defined by evaluating existing documentation on the storm as well as plotting and evaluating initial precipitation gauge data on a map. The analysis domain is defined to include as many hourly recording gauges as possible given their importance in timing. The domain must include enough of a buffer to accurately model the nested domain of interest. The domain is defined as a longitude-latitude (upper left and lower right corner) rectangular region.

SPAS Analysis Time Frame

Ideally, the analysis time frame, also referred to as the Storm Precipitation Period (SPP), will extend from a dry period through the target wet period then back into another dry period. This is to ensure that total storm precipitation amounts can be confidently associated with the storm in question and not contaminated by adjacent wet periods. If this is not possible, a reasonable time

period is selected that is bounded by relatively lighter precipitation. The time frame of the hourly data must be sufficient to capture the full range of daily gauge observational periods for the daily observations to be disaggregated into estimated incremental hourly values (discussed later). For example, if a daily gauge takes observations at 8:00 AM, then the hourly data must be available from 8:00 AM the day prior. Given the configuration of SPAS, the minimum SPP is 72 hours and aligns midnight to midnight.
The core precipitation period (CPP) is a sub-set of the SPP and represents the time period with the most precipitation and the greatest number of reporting gauges. The CPP represents the time period of interest and where our confidence in the results is highest.

Data

The foundation of a SPAS analysis is the "ground truth" precipitation measurements. In fact, the level of effort involved in "data mining" and quality control represent over half of the total level of effort needed to conduct a complete storm analysis. SPAS operates with three primary data sets: precipitation gauge data, a basemap and, if available, radar data. Table E.1 conveys the variety of precipitation gauges usable by SPAS. For each gauge, the following elements are gathered, entered and archived into SPAS database:

- Station ID
- Station name
- Station type (H=hourly, D=Daily, S=Supplemental, etc.)
- Longitude in decimal degrees
- Latitude in decimal degrees
- Elevation in feet above MSL
- Observed precipitation
- Observation times
- Source
- If unofficial, the measurement equipment and/or method is also noted.

Based on the SPP and analysis domain, hourly and daily precipitation gauge data are extracted from our in-house database as well as the Meteorological Assimilation Data Ingest System (MADIS). Our in-house database contains data dating back to the late 1800s, while the MADIS system (described below) contains archived data back to 2002.

Hourly Precipitation Data

Our hourly precipitation database is largely comprised of data from NCDC TD-3240, but also precipitation data from other mesonets and meteorological networks (e.g., ALERT, Flood Control Districts, etc.) that we have collected and archived as part of previous studies. Meanwhile, MADIS provides data from a large number of networks across the U.S., including NOAA's HADS (Hydrometeorological Automated Data System), numerous mesonets, the Citizen Weather Observers Program (CWOP), departments of transportation, etc. (see http://madis.noaa.gov/mesonet_providers.html for a list of providers). Although our automatic data extraction is fast, cost-effective and efficient, it never captures all of the available precipitation data for a storm event. For this reason, a thorough "data mining" effort is undertaken to acquire all available data from sources such as U.S. Geological Survey (USGS), Remote Automated Weather Stations (RAWS), Community Collaborative Rain, Hail & Snow Network (CoCoRaHS), National Atmospheric Deposition Program (NADP), Clean Air Status and Trends Network (CASTNET), local observer networks, Climate Reference Network (CRN), Global Summary of the Day (GSD) and Soil Climate Analysis Network (SCAN). Unofficial hourly precipitation data are gathered to give guidance on either timing or magnitude in areas otherwise void of precipitation data. The WeatherUnderground and MesoWest, two of the largest weather databases on the Internet, contain a large proportion of official data, but also includes data from unofficial gauges.

Table E.1: Different precipitation gauge types used by SPAS

Precipitation Gauge Type	Description
Hourly	Hourly gauges with complete, or nearly complete, incremental hourly precipitation data.
Hourly estimated	Hourly gauges with some estimated hourly values, but otherwise reliable.
Hourly pseudo	Hourly gauges with reliable temporal precipitation data, but the magnitude is questionable in relation to co-located daily or supplemental gauge.
Daily	Daily gauge with complete data and known observation times.
Daily estimated	Daily gauges with some or all estimated data.
Supplemental	Gauges with unknown or irregular observation times, but reliable total storm precipitation data. (E.g. public reports, storms reports, "Bucket surveys", etc.)
Supplemental estimated	Gauges with estimated total storm precipitation values based on other information (e.g. newspaper articles, stream flow discharge, inferences from nearby gauges, pre-existing total storm isohyetal maps, etc.)

Daily Precipitation Data

Our daily database is largely based on NCDC's TD-3206 (pre-1948) and TD-3200 (1948 through present) as well as SNOTEL data from NRCS. Since the late 1990s, the CoCoRaHS network of more than 15,000 observers in the U.S. has become a very important daily precipitation source. Other daily data are gathered from similar, but smaller gauge networks, for instance the High Spatial Density Precipitation Network in Minnesota.

As part of the daily data extraction process, the time of observation accompanies each measured precipitation value. Accurate observation times are necessary for SPAS to disaggregate the daily precipitation into estimated incremental values (discussed later). Knowing the observation time also allows SPAS to maintain precipitation amounts within given time bounds, thereby retaining known precipitation intensities. Given the importance of observation times, efforts are taken to mase sure the observation times are accurate. Hardcopy reports of "Climatological Data," scanned observational forms (available on-line from the NCDC) and/or gauge metadata forms have proven to be valuable and accurate resources for validating observation times. Furthermore, erroneous observation times are identified in the mass-curve quality-control procedure (discussed later) and can be corrected at that point in the process.

Supplemental Precipitation Gauge Data

For gauges with unknown or irregular observation times, the gauge is considered a "supplemental" gauge. A supplemental gauge can either be added to the storm database with a storm total and the associated SPP as the temporal bounds or as a gauge with the known, but irregular observation times and associated precipitation amounts. For instance, if all that is known is 3 inches fell between 0800-0900, then that information can be entered. Gauges or reports with nothing more than a storm total are often abundant, but to use them, it is important the precipitation is only from the storm period in question. Therefore, it is ideal to have the analysis time frame bounded by dry periods.

Perhaps the most important source of data, if available, is from "bucket surveys," which provide comprehensive lists of precipitation measurements collected during a post-storm field exercise. Although some bucket survey amounts are not from conventional precipitation gauges, they provide important information, especially in areas lacking data. Particularly for PMP-storm analysis applications, it is customary to accept extreme, but valid non-standard precipitation values (such as bottles and other open containers that catch rainfall) to capture the highest precipitation values.

Basemap

"Basemaps" are independent grids of spatially distributed weather or climate variables that are used to govern the spatial patterns of the hourly precipitation. The basemap also governs the spatial resolution of the final SPAS grids unless radar data are available/used to govern the spatial resolution. Note that a base map is not required as the hourly precipitation patterns can be based on station characteristics and an inverse distance weighting technique (discussed later). Basemaps in complex terrain are often based on the PRISM mean monthly precipitation (Figure E.2a) or Hydrometeorological Design Studies Center precipitation frequency grids (Figure E.2b) given they resolve orographic enhancement areas and micro-climates at a spatial resolution of 30-seconds (about 800 m). Basemaps of this nature in flat terrain are not as effective given the small terrain forced precipitation gradients. Therefore, basemaps for SPAS analyses in flat terrain are often developed from pre-existing (hand-drawn) isohyetal patterns (Figure E.2c), composite radar imagery or a blend of both.

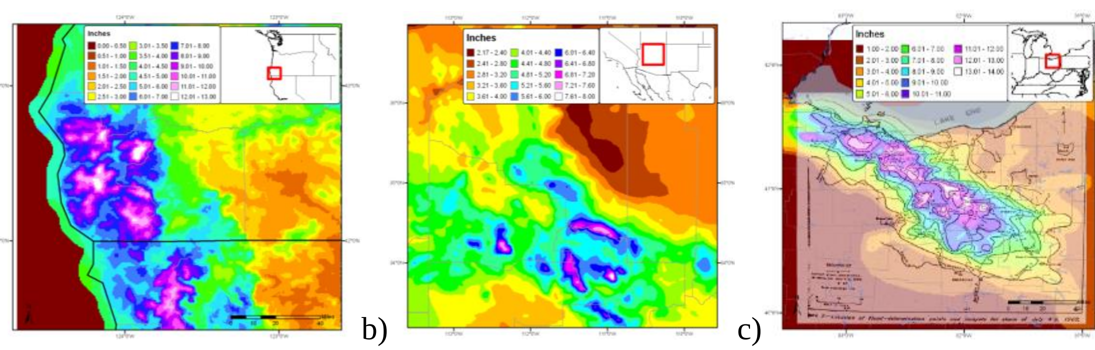

a) b) c)

Figure E.2: Sample SPAS "basemaps:" (a) A pre-existing (USGS) isohyetal pattern across flat terrain (SPAS #1209), (b) PRISM mean monthly (October) precipitation (SPAS #1192) and (c) A 100-year 24-hour precipitation grid from NOAA Atlas 14 (SPAS #1138)

Radar Data

For storms occurring since approximately the mid-1990s, weather radar data are available to supplement the SPAS analysis. A fundamental requirement for high quality radar-estimated precipitation is a high quality radar mosaic, which is a seamless collection of concurrent weather radar data from individual radar sites, however in some cases a single radar is sufficient (i.e. for a small area size storm event such as a thunderstorm). Weather radar data have been in use by meteorologists since the 1960s to estimate precipitation depths, but it was not until the early 1990s that new, more accurate NEXRAD Doppler radar (WSR88D) was placed into service across the United States. Currently, efforts are underway to convert the WSR88D radars to dual polarization (DualPol) radar. Today, NEXRAD radar coverage of the contiguous United States is comprised of 159 operational sites and there are 30 in Canada. Each U.S. radar covers an approximate 285 mile (460 km) radial extent while Canadian radars have approximately 256 km (138 nautical miles) radial extent over which their radar can detect precipitation (see Figure E.3). The primary vendor of NEXRAD weather radar data for SPAS is Weather Decision Technologies, Inc. (WDT), who accesses, mosaics, archives and quality-controls NEXRAD radar data from NOAA and Environment Canada. SPAS utilizes Level II NEXRAD radar reflectivity data in units of dBZ, available every 5-minutes in the U.S. and 10-minutes in Canada.

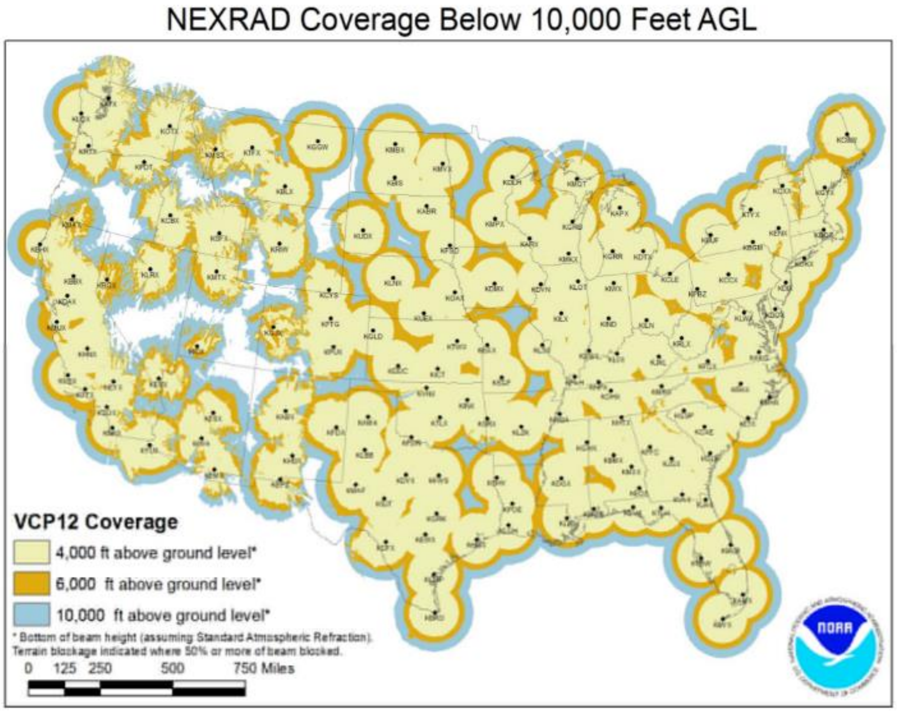


Figure E.3: U.S. radar locations and their radial extents of coverage below 10,000 feet above ground level (AGL). Each U.S. radar covers an approximate 285 mile radial extent over which the radar can detect precipitation.

The WDT and National Severe Storms Lab (NSSL) Radar Data Quality Control Algorithm (RDQC) removes non-precipitation artifacts from base Level–II radar data and remaps the data from polar coordinates to a Cartesian (latitude/longitude) grid. Non-precipitation artifacts include ground clutter, bright banding, sea clutter, anomalous propagation, sun strobes, clear air returns, chaff, biological targets, and electronic interference and hardware test patterns. The RDQC algorithm uses sophisticated data processing and a Quality Control Neural Network (QCNN) to delineate the precipitation echoes caused by radar artifacts (Lakshmanan and Valente 2004). Beam blockages due to terrain are mitigated by using 30-meter DEM data to compute and then discard data from a radar beam that clears the ground by less than 50 meters and incurs more than 50% power blockage. A clear-air echo removal scheme is applied to radars in clear-air mode when there is no precipitation reported from observation gauges within the vicinity of the radar. In areas of radar coverage overlap, a distance weighting scheme is applied to assign reflectivity to each grid cell, for multiple vertical levels. This scheme is applied to data from the nearest radar that is unblocked by terrain.

Once data from individual radars have passed through the RDQC, they are merged to create a seamless mosaic for the United States and southern Canada as shown in Figure E.4. A multi-sensor quality control can be applied by post-processing the mosaic to remove any remaining "false echoes." This technique uses observations of infra-red cloud top temperatures by GOES satellite and surface temperature to create a precipitation/no-precipitation mask. Figure E.4(b) shows the impact of WDT's quality control measures. Upon completing all QC, WDT converts the radar data from its native polar coordinate projection (1 degree x 1.0 km) into a longitude-latitude Cartesian grid (based on the WGS84 datum), at a spatial resolution of ~1/3rdmi^2 for processing in SPAS.

a) b)

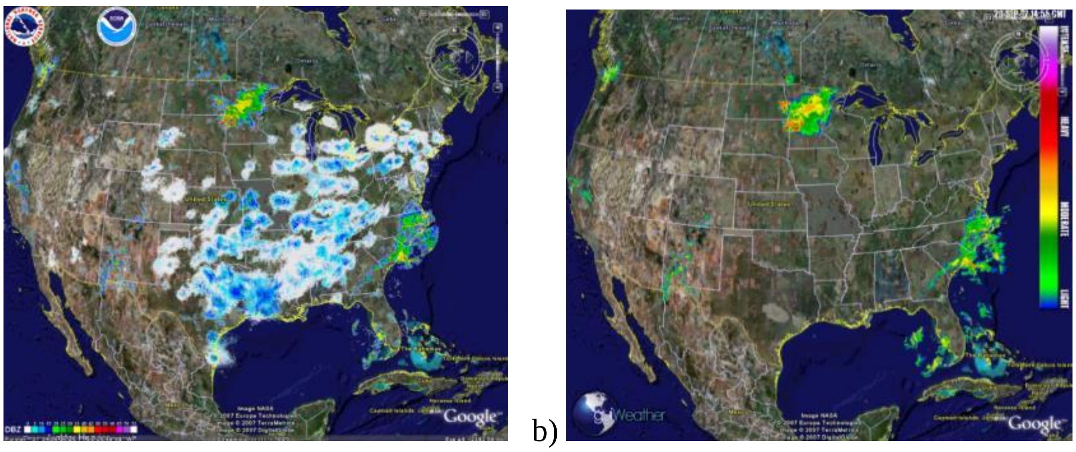


Figure E.4: (a) Level-II radar mosaic of CONUS radar with no quality control, (b) WDT quality controlled Level-II radar mosaic

SPAS conducts further QC on the radar mosaic by infilling areas contaminated by beam blockages. Beam blocked areas are objectively determined by evaluating total storm reflectivity grid which naturally amplifies areas of the SPAS analysis domain suffering from beam blockage as shown in Figure E.5.

a) b)

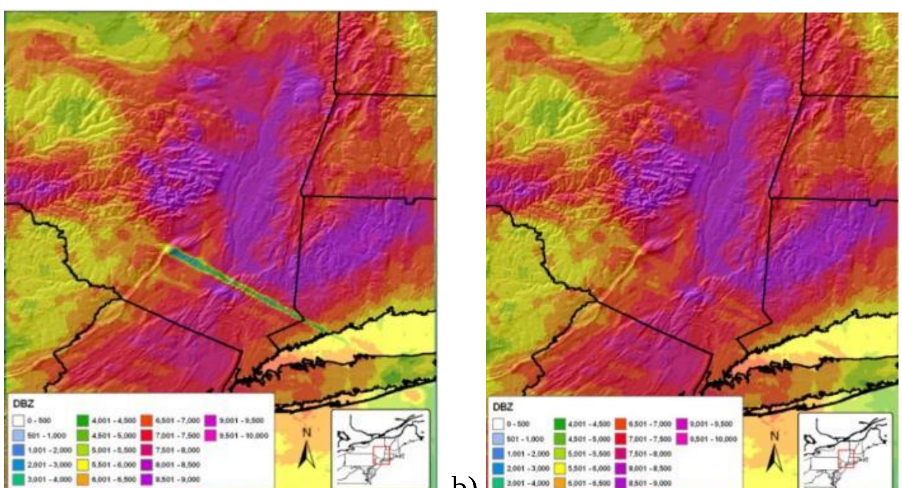


Figure E.5: Illustration of SPAS-beam blockage infilling where (a) is raw, blocked radar and (b) is filled for a 42-hour storm event

Methodology

Daily and Supplemental Precipitation to Hourly

To obtain one hour temporal resolutions and utilize all gauge data, it is necessary to disaggregate daily and supplemental precipitation observations into estimated hourly amounts. This process has traditionally been accomplished by distributing (temporally) the precipitation at each daily/supplemental gauge in accordance to a single nearby hourly gauge (Thiessen polygon approach). However, this may introduce biases and not correctly represent hourly precipitation at daily/supplemental gauges situated in-between hourly gauges. Instead, SPAS uses a spatial approach by which the estimated hourly precipitation at each daily and supplemental gauge is governed by a distance weighted algorithm of all nearby true hourly gauges.

To disaggregate (i.e., distribute) daily/supplemental gauge data into estimate hourly values, the true hourly gauge data are first evaluated, and quality controlled using synoptic maps, nearby gauges, orographic effects, gauge history and other documentation on the storm. Any problems with the hourly data are resolved, and when possible/necessary accumulated hourly values are distributed. If an hourly value is missing, the analyst can choose to either estimate it or leave it missing for SPAS to estimate later based on nearby hourly gauges. At this point in the process, pseudo (hourly) gauges can be added to represent precipitation timing in topographically complex locations, areas with limited/no hourly data or to capture localized convention. Hourly Pseudo stations add additional detail on the timing of rainfall, either from COOP forms, radar reflectivity timing, and/or bucket survey reports with time increments. Hourly Pseudo stations are used only for the timing surrounding daily and supplemental stations and not for the magnitude. The limitations of Hourly Pseudo stations are that they are based on surrogate information, the quality of the information can be highly questionable (based on source) thus the importance of the station QC procedures are extremely important. To adequately capture the temporal variations of the precipitation, a pseudo hourly gauge is sometimes necessary. A pseudo gauge is created by distributing the precipitation at a co-located daily gauge or by creating a completely new pseudo gauge from other information such as inferences from COOP observation forms, METAR visibility data (if hourly precipitation is not already available), lightning data, satellite data, or radar data. Often radar data are the best/only choice for creating pseudo hourly gauges, but this is done cautiously given the potential differences (over-shooting of the radar beam equating to erroneous precipitation) between radar data and precipitation. In any case, the pseudo hourly gauge is flagged so SPAS only uses it for timing and not magnitude. Care is taken to ensure hourly pseudo gauges represent justifiably important physical and meteorological characteristics before being incorporated into the SPAS database. Although pseudo gauges provide a very important role, their use is kept to a minimum. The importance of having accurate reliability of every hourly gauge cannot be over emphasized. All of the final hourly gauge data, including pseudos, are included in the hourly SPAS precipitation database.

Using the hourly SPAS precipitation database, each hourly precipitation value is converted into a percentage that represents the incremental hourly precipitation divided by the total SPP precipitation. The GIS-ready x-y-z file is constructed for each hour and it includes the latitude (x), longitude(y) and the percent of precipitation (z) for a particular hour. Using the GRASS GIS, an inverse-distance-weighting squared (IDW) interpolation technique is applied to each of the hourly files. The result is a continuous grid with percentage values for the entire analysis domain, keeping the grid cells on which the hourly gauge resides faithful to the observed/actual percentage. Since the percentages typically have a high degree of spatial autocorrelation, the spatial interpolation has skill in determining the percentages between gauges, especially since the percentages are somewhat independent of the precipitation magnitude. The end result is a GIS grid for each hour that represents the percentage of the SPP precipitation that fell during that hour.

After the hourly percentage grids are generated and QC'd for the entire SPP, a program is executed that converts the daily/supplemental gauge data into incremental hourly data. The timing at each of the daily/supplemental gauges is based on (1) the daily/supplemental gauge observation time, (2) daily/supplemental precipitation amount and (3) the series of interpolated hourly percentages extracted from grids (described above).

This procedure is detailed in Figure E.6 below. In this example, a supplemental gauge reported 1.40" of precipitation during the storm event and is located equal distance from the three surrounding hourly recording gauges. The procedure steps are:

Step 1. For each hour, extract the percent of SPP from the hourly gauge-based percentage at the location of the daily/supplemental gauge. In this example, assume these values are the average of all the hourly gauges.

Step 2. Multiply the individual hourly percentages by the total storm precipitation at the daily/supplemental gauge to arrive at estimated hourly precipitation at the daily/supplemental gauge. To make the daily/supplemental accumulated precipitation data faithful to the daily/supplemental observations, it is sometimes necessary to adjust the hourly percentages so they add up to 100% and account for 100% of the daily observed precipitation.

	Hour						
Precipitation	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
Hourly station 1	0.02	0.12	0.42	0.50	0.10	0.00	1.16
Hourly station 2	0.01	0.15	0.48	0.62	0.05	0.01	1.32
Hourly station 3	0.00	0.18	0.38	0.55	0.20	0.05	1.36
	Hour						
Percent of total storm precip.	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
Hourly station 1	2%	10%	36%	43%	9%	0%	100%
Hourly station 2	1%	11%	36%	47%	4%	1%	100%
Hourly station 3	0%	13%	28%	40%	15%	4%	100%
Average	*1%*	*12%*	*34%*	*44%*	*9%*	*1%*	*100%*
Storm total precipitation at daily gauge				1.40			
	Hour						
Precipitation (estimated)	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
Daily station	0.01	0.16	0.47	0.61	0.13	0.02	1.40

Figure E.6: Example of disaggregation of daily precipitation into estimated hourly precipitation based on three (3) surrounding hourly recording gauges

In cases where the hourly grids do not indicate any precipitation falling during the daily/supplemental gauge observational period, yet the daily/supplemental gauge reported precipitation, the daily/supplemental total precipitation is evenly distributed throughout the hours that make up the observational period; although this does not happen very often, this solution is consistent with NWS procedures. However, the SPAS analyst is notified of these cases in a comprehensive log file, and in most cases, they are resolvable, sometimes with a pseudo hourly gauge.

Gauge Quality Control

Exhaustive quality control measures are taken throughout the SPAS analysis. Below are a few of the most significant QC measures taken.

Mass Curve Check

A mass curve-based QC-methodology is used to ensure the timing of precipitation at all gauges is consistent with nearby gauges. SPAS groups each gauge with the nearest four gauges (regardless of type) into a single file. These files are subsequently used in software for graphing and evaluation. Unusual characteristics in the mass curve are investigated and the gauge data corrected, if possible and warranted. See Figure E.7 for an example.

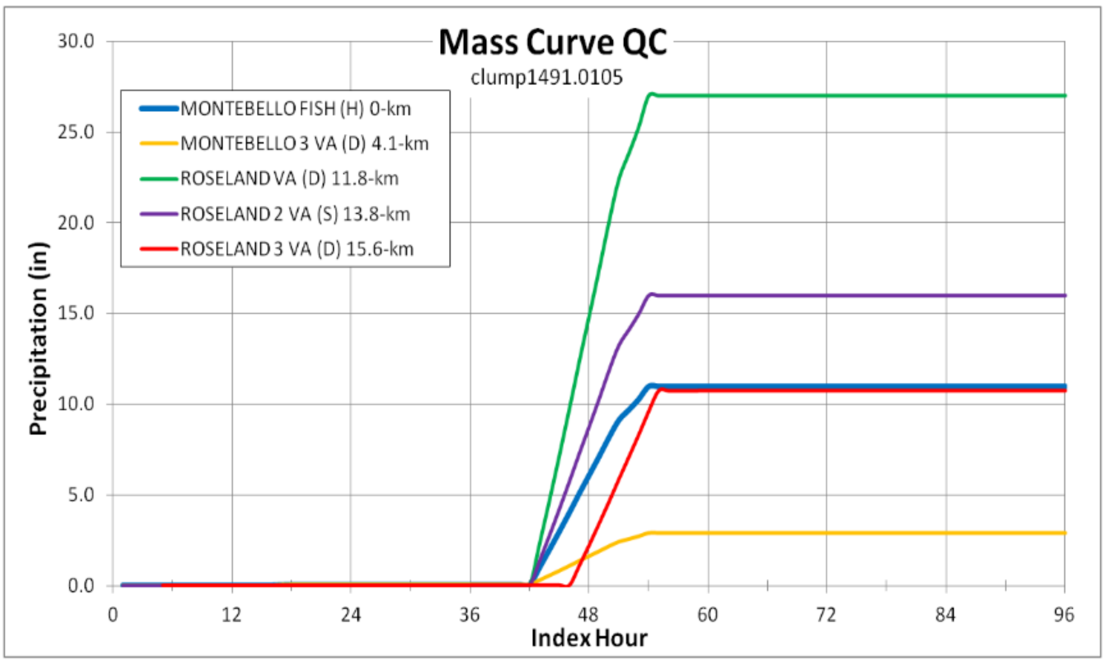


Figure E.7: Sample mass curve plot depicting a precipitation gauge with an erroneous observation time (red line). X-axis is the SPAS index hour and the y-axis is inches. The statistics in the upper left denote gauge type, and distance from target gauge (in km). In this example, the daily gauge (red line) was found to have an observation error/shift of 6-hours.

Gauge Mis-location Check

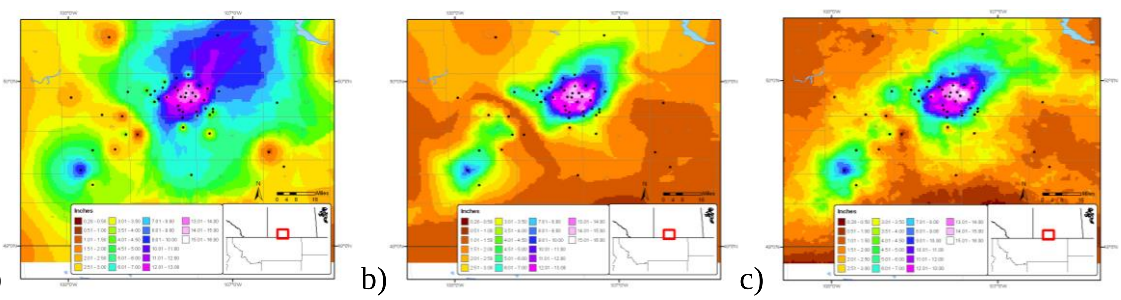
Although the gauge elevation is not explicitly used in SPAS, it is however used as a means of QC'ing gauge location. Gauge elevations are compared to a high-resolution 15-second DEM to identify gauges with large differences, which may indicate erroneous longitude and/or latitude values.

Co-located Gauge QC

Care is also taken to establish the most accurate precipitation depths at all co-located gauges. In general, where a co-located gauge pair exists, the highest precipitation is accepted (if deemed accurate). If the hourly gauge reports higher precipitation, then the co-located daily (or supplemental) is removed from the analysis since it would not add anything to the analysis. Often daily (or supplemental) gauges report greater precipitation than a co-located hourly station since hourly tipping bucket gauges tend to suffer from gauge under-catch, particularly during extreme events, due to loss of precipitation during tips. In these cases, the daily/supplemental is retained for the magnitude and the hourly used as a pseudo hourly gauge for timing. Large discrepancies between any co-located gauges are investigated and resolved since SPAS can only utilize a single gauge magnitude at each co-located site.

Spatial Interpolation

At this point the QC'd observed hourly and disaggregated daily/supplemental hourly precipitation data are spatially interpolated into hourly precipitation grids. SPAS has three options for conducting the hourly precipitation interpolation, depending on the terrain and availability of radar data, thereby allowing SPAS to be optimized for any particular storm type or location. Figure E.8 depicts the results of each spatial interpolation methodology based on the same precipitation gauge data.



a) b) c)

Figure E.8: Depictions of total storm precipitation based on the three SPAS interpolation methodologies for a storm (SPAS #1177, Vanguard, Canada) across flat terrain: (a) no basemap, (b) basemap-aided and (c) radar

Basic Approach

The basic approach interpolates the hourly precipitation point values to a grid using an inverse distance weighting squared GIS algorithm. This is sometimes the best choice for convective storms over flat terrain when radar data are not available, yet high gauge density instills reliable precipitation patterns. This approach is rarely used.

Basemap Approach

Another option includes use of a basemap, also known as a climatologically-aided interpolation (Hunter 2005). As noted before, the spatial patterns of the basemap govern the interpolation between points of hourly precipitation estimates, while the actual hourly precipitation values govern the magnitude. This approach to interpolating point data across complex terrain is widely used. In fact, it was used extensively by the NWS during their storm analysis era from the 1940s through the 1970s (USACE 1973, Hansen et al., 1988, Corrigan et al., 1999).

In application, the hourly precipitation gauge values are first normalized by the corresponding grid cell value of the basemap before being interpolated. The normalization allows information and knowledge from the basemap to be transferred to the spatial distribution of the hourly precipitation. Using an IDW squared algorithm, the normalized hourly precipitation values are interpolated to a grid. The resulting grid is then multiplied by the basemap grid to produce the hourly precipitation grid. This is repeated each hour of the storm.

Radar Approach

The coupling of SPAS with NEXRAD provides the most accurate method of spatially and temporally distributing precipitation. To increase the accuracy of the results however, quality-controlled precipitation observations are used for calibrating the radar reflectivity to rain rate relationship (Z-R relationship) each hour instead of assuming a default Z-R relationship. Also, spatial variability in the Z-R relationship is accounted for through local bias corrections (described later). The radar approach involves several steps, each briefly described below. The radar approach cannot operate alone – either the basic or basemap approach must be completed before radar data can be incorporated. The SPAS general code is where the daily and supplemental station are timed to hourly data. Therefore, to get the correct timing of daily and supplemental stations, SPAS general needs to be run. The timed hourly data are used as input into SPAS-NEXRAD to derive the dynamic ZR relationship each hour.

Basemaps are only used to aid in the spatial interpolation. In regards to SPAS-NEXRAD, a basemap is used to interpolate the radar residuals (bias adjustments).

Z-R Relationship

SPAS derives high quality precipitation estimates by relating quality controlled level–II NEXRAD radar reflectivity radar data with quality-controlled precipitation gauge data to calibrate the Z-R (radar reflectivity, Z, and precipitation, R) relationship. Optimizing the Z-R relationship is essential for capturing temporal changes in the Z-R. Most current radar-derived precipitation techniques rely on a constant relationship between radar reflectivity and precipitation rate for a given storm type (e.g., tropical, convective), vertical structure of reflectivity and/or reflectivity magnitudes. This non-linear relationship is described by the Z-R equation below:

$$Z = A\ R^{b} \qquad (1)$$

Where Z is the radar reflectivity (measured in units of dBZ), R is the precipitation (precipitation) rate (millimeters per hour), A is the “multiplicative coefficient” and b is the “power coefficient”. Both A and b are directly related to the rain drop size distribution (DSD) and rain drop number distribution (DND) within a cloud (Martner and Dubovskiy 2005). The variability in the results of Z versus R is a direct result of differing DSD, DND and air mass characteristics (Dickens 2003). The DSD and DND are determined by complex interactions of microphysical processes that fluctuate regionally, seasonally, daily, hourly, and even within the same cloud. For these reasons, SPAS calculates an optimized Z-R relationship across the analysis domain each hour, based on observed precipitation rates and radar reflectivity (see Figure E.9).

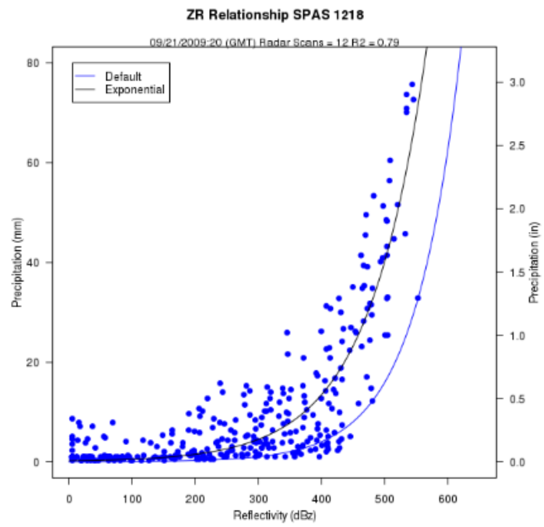


Figure E.9: Example SPAS (denoted as "Exponential") vs. default Z-R relationship (SPAS #1218, Georgia September 2009)

The National Weather Service (NWS) utilizes different default Z-R algorithms, depending on the type of precipitation event, to estimate precipitation from NEXRAD radar reflectivity data across the United States (see Figure E.10) (Baeck and Smith 1998 and Hunter 1999). A default Z-R relationship of Z = 300R1.4 is the primary algorithm used throughout the continental U.S. However, it is widely known that this, compared to unadjusted radar-aided estimates of precipitation, suffers from deficiencies that may lead to significant over or under-estimation of precipitation.

RELATIONSHIP	Optimum for:	Also recommended for:
Marshall-Palmer (z=*200R$^{1.6}$*)	General stratiform precipitation	
East-Cool Stratiform (z=*130R$^{2.0}$*)	Winter stratiform precipitation - east of continental divide	Orographic rain - East
West-Cool Stratiform (z=*75R$^{2.0}$*)	Winter stratiform precipitation - west of continental divide	Orographic rain - West
WSR-88D Convective (z=*300R$^{1.4}$*)	Summer deep convection	Other non-tropical convection
Rosenfeld Tropical (z=*250R$^{1.2}$*)	Tropical convective systems	

Figure E.10: Commonly used Z-R algorithms used by the NWS

Instead of adopting a standard Z-R, SPAS utilizes a least squares fit procedure for optimizing the Z-R relationship each hour of the SPP. The process begins by determining if sufficient (minimum 12) observed hourly precipitation and radar data pairs are available to compute a reliable Z-R. If insufficient (<12) gauge pairs are available, then SPAS adopts the previous hour Z-R relationship, if available, or applies a user-defined default Z-R algorithm. If sufficient data are available, the one-hour sum of NEXRAD reflectivity (Z) is related to the 1-hour precipitation at each gauge. A least-squares-fit exponential function using the data points is computed. The resulting best-fit, one hour-based Z-R is subjected to several tests to determine if the Z-R relationship and its resulting precipitation rates are within a certain tolerance based on the R-squared fit measure and difference between the derived and default Z-R precipitation results. Experience has shown the actual Z-R versus the default Z-R can be significantly different (Figure E.11). These Z-R relationships vary by storm type and location. A standard output of all SPAS analyses utilizing NEXRAD includes a file with each hour's adjusted Z-R relationship as calculated through the SPAS program.

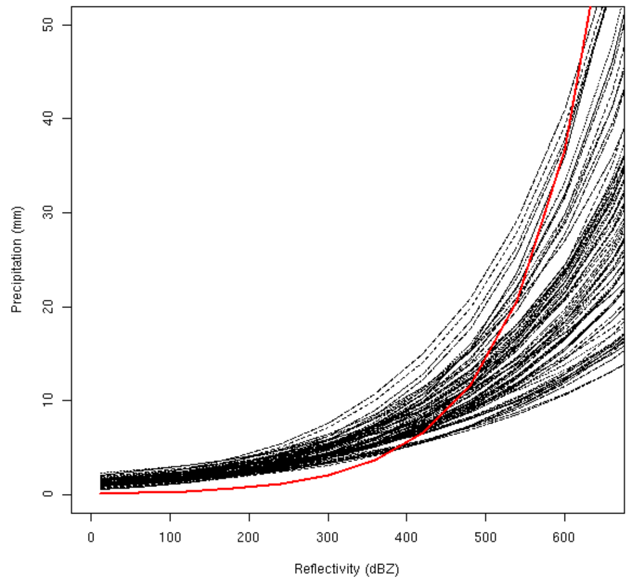


Figure E.11: Comparison of the SPAS optimized hourly Z-R relationships (black lines) versus a default Z=75R2.0 Z-R relationship (red line) for a period of 99 hours for a storm over southern California.

Radar-aided Hourly Precipitation Grids

Once a mathematically optimized hourly Z-R relationship is determined, it is applied to the total hourly Z grid to compute an initial precipitation rate (inches/hour) at each grid cell. To account for spatial differences in the Z-R relationship, SPAS computes residuals, the difference between the initial precipitation analysis (via the Z-R equation) and the actual “ground truth” precipitation (observed – initial analysis), at each gauge. The point residuals, also referred to as local biases, are normalized and interpolated to a residual grid using an inverse distance squared weighting algorithm. A radar-based hourly precipitation grid is created by adding the residual grid to the initial grid; this allows precipitation at the grid cells for which gauges are “on” to be true and faithful to the gauge measurement. The pre-final radar-aided precipitation grid is subject to some final, visual QC checks to ensure the precipitation patterns are consistent with the terrain; these checks are particularly important in areas of complex terrain where even QC’d radar data can be unreliable. The next incremental improvement with SPAS program will come as the NEXRAD radar sites are upgraded to dual-polarimetric capability.

Radar- and Basemap-Aided Hourly Precipitation Grids

At this stage of the radar approach, a radar- and basemap-aided hourly precipitation grid exists for each hour. At locations with precipitation gauges, the grids are equal, however elsewhere the grids can vary for a number of reasons. For instance, the basemap-aided hourly precipitation grid may depict heavy precipitation in an area of complex terrain, blocked by the radar, whereas the radar-aided hourly precipitation grid may suggest little, if any, precipitation fell in the same area. Similarly, the radar-aided hourly precipitation grid may depict an area of heavy precipitation in flat terrain that the basemap-approach missed since the area of heavy precipitation occurred in an area without gauges. SPAS uses an algorithm to compute the hourly precipitation at each pixel given the two results. Areas that are completely blocked from a radar signal are accounted for with the basemap-aided results (discussed earlier). Precipitation in areas with orographically effective terrain and reliable radar data are governed by a blend of the basemap- and radar-aided precipitation. Elsewhere, the radar-aided precipitation is used exclusively. This blended approach has proven effective for resolving precipitation in complex terrain yet retaining accurate radar-aided precipitation across areas where radar data are reliable. Figure E.12 illustrates the evolution of final precipitation from radar reflectivity in an area of complex terrain in southern California.

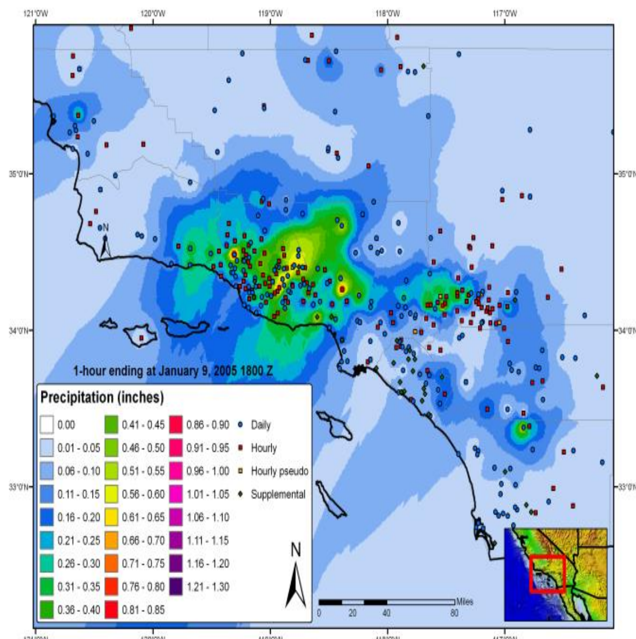


Figure E.12a: Map depicting 1-hour of precipitation utilizing inverse distance weighting of gauge precipitation for a January 2005 storm in southern California, USA

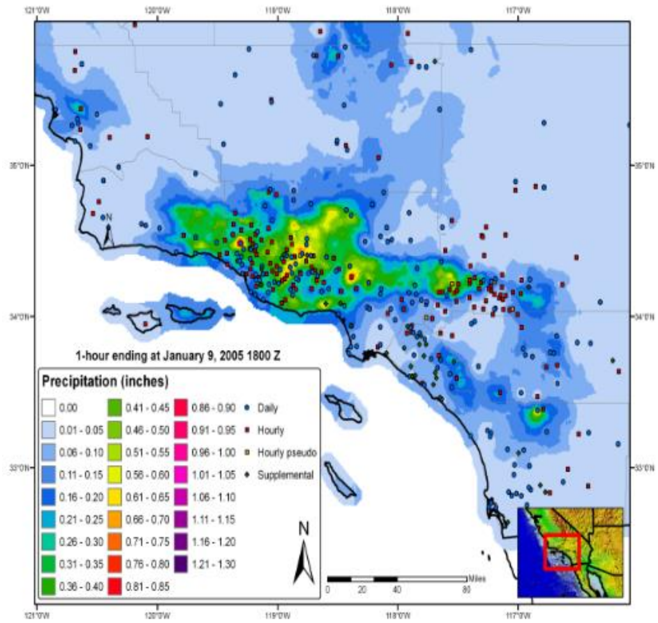


Figure E.12b: Map depicting 1-hour of precipitation utilizing gauge data together with a climatologically-aided interpolation scheme for a January 2005 storm in southern California, USA

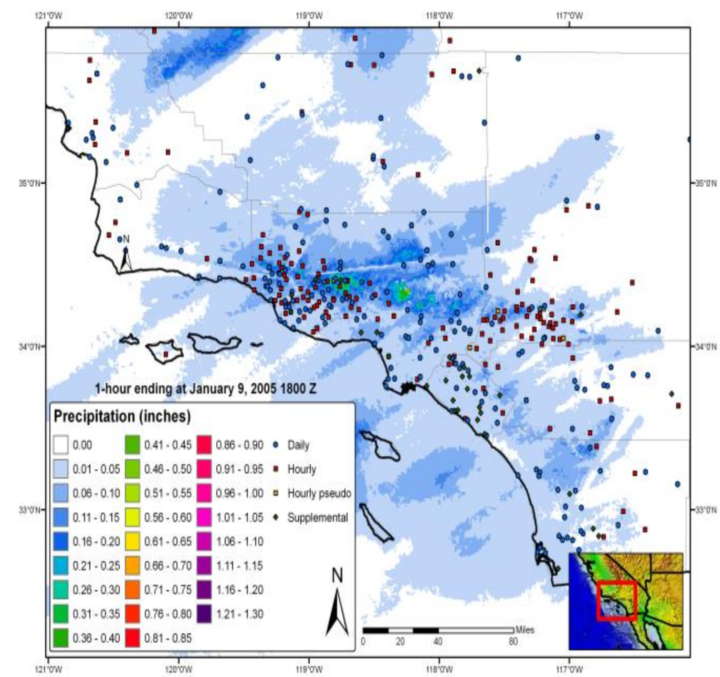


Figure E.12c: Map depicting 1-hour of precipitation utilizing default Z-R radar-estimated interpolation (no gauge correction) for a January 2005 storm in southern California, USA

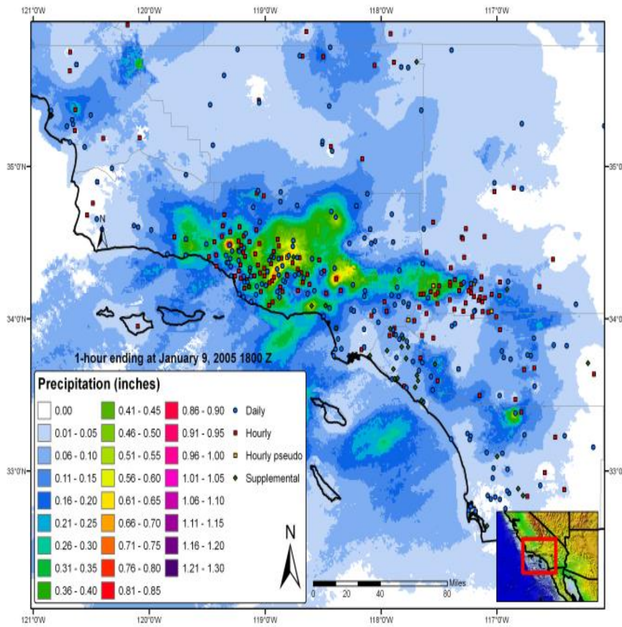


Figure E.12d: Map depicting 1-hour of precipitation utilizing SPAS precipitation for a January 2005 storm in southern California, USA

SPAS versus Gauge Precipitation

Performance measures are computed and evaluated each hour to detect errors and inconsistencies in the analysis. The measures include hourly Z-R coefficients, observed hourly maximum precipitation, maximum gridded precipitation, hourly bias, hourly mean absolute error (MAE), root mean square error (RMSE), and hourly coefficient of determination (r^2).

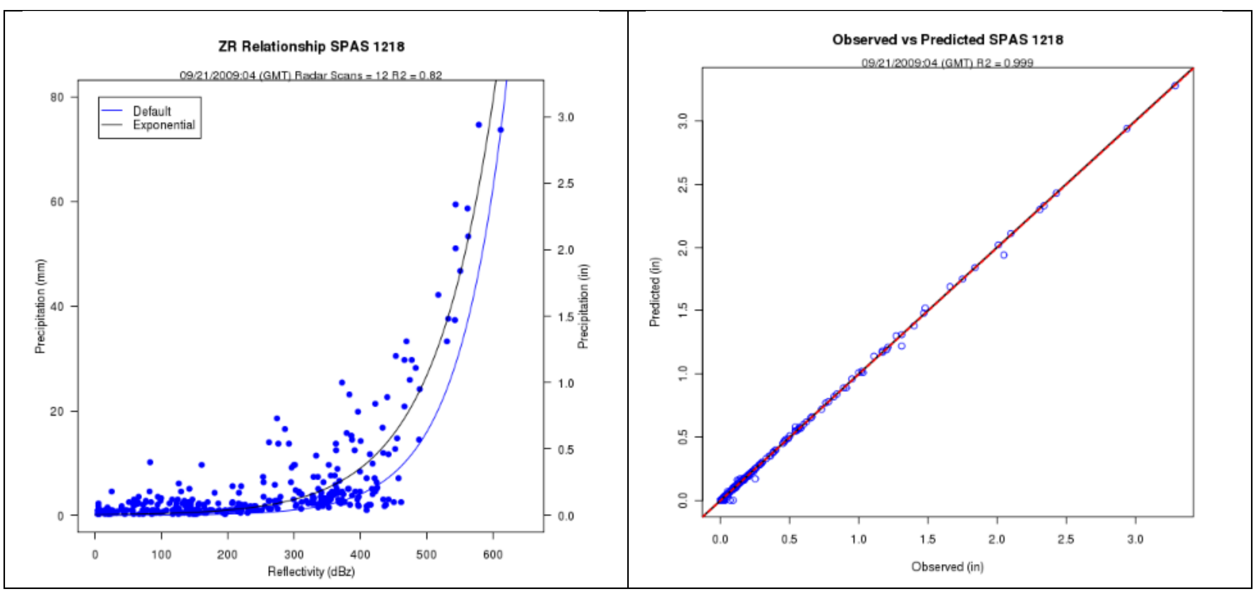


Figure E.13: Z-R plot (a), where the blue line is the SPAS derived Z-R and the black line is the default Z-R, and the (b) associated observed versus SPAS scatter plot at gauge locations.

Comparing SPAS-calculated precipitation (R_{spas}) to observed point precipitation depths at the gauge locations provides an objective measure of the consistency, accuracy and bias. Generally speaking SPAS is usually within 5% of the observed precipitation (see Figure E.13). Less-than-perfect correlations between SPAS precipitation depths and observed precipitation at gauged locations could be the result of any number of issues, including:

- **Point versus area:** A rain gauge observation represents a much smaller area than the area sampled by the radar. The area that the radar is sampling is approximately 1 km^2, whereas a standard rain gauge has an opening 8 inches in diameter, hence it only samples approximately 8.0×10^{-9} km^2. Furthermore, the radar data represent an average reflectivity (Z) over the grid cell, when in fact the reflectivity can vary across the 1 km^2 grid cell. Therefore, comparing a grid cell radar derived precipitation value to a gauge (point) precipitation depth measured may vary.
- **Precipitation gauge under-catch:** Although we consider gauge data "ground truth," we recognize gauges themselves suffer from inaccuracies. Precipitation gauges, shielded and unshielded, inherently underestimate total precipitation due to local airflow, wind under-catch, wetting, and evaporation. The wind under-catch errors are usually around 5% but can be as large as 40% in high winds (Guo et al., 2001, Duchon and Essenberg 2001, Ciach 2003, Tokay et al., 2010). Tipping buckets miss a small amount of precipitation during each tip of the bucket due to the bucket travel and tip time. As precipitation intensities increase, the volumetric loss of precipitation due to tipping tends to increase. Smaller tipping buckets can have higher volumetric losses due to higher tip frequencies, but on the other hand capture higher precision timing.
- **Radar Calibration:** NEXRAD radars calibrate reflectivity every volume scan, using an internally generated test. The test determines changes in internal variables such as beam power and path loss of the receiver signal processor since the last off-line calibration. If this value becomes large, it is likely that there is a radar calibration error that will translate into less reliable precipitation estimates. The calibration test is supposed to maintain a reflectivity precision of 1 dBZ. A 1 dBZ error can result in an error of up to 17% in R_{spas} using the default Z-R relationship $Z=300R^{1.4}$. Higher calibration errors will result in higher R_{spas} errors. However, by performing correlations each hour, the calibration issue is minimized in SPAS.
- **Attenuation:** Attenuation is the reduction in power of the radar beams' energy as it travels from the antenna to the target and back. It is caused by the absorption and the scattering of power from the beam by precipitation. Attenuation can result in errors in Z as large as 1 dBZ especially when the radar beam is sampling a large area of heavy precipitation. In some cases, storm precipitation is so intense (>12 inches/hour) that individual storm cells become "opaque" and the radar beam is totally attenuated. Armed with sufficient gauge data however, SPAS will overcome attenuation issues.
- **Range effects:** The curvature of Earth and radar beam refraction result in the radar beam becoming more elevated above the surface with increasing range. With the increased elevation of the radar beam comes a decrease in Z values due to the radar beam not sampling the main precipitation portion of the cloud (i.e., "over topping" the precipitation and/or cloud altogether). Additionally, as the radar beam gets further from the radar, it naturally samples a larger and larger area, therefore amplifying point versus area differences (described above).
- **Radar Beam Occultation/Ground Clutter:** Radar occultation (beam blockage) results when the radar beam's energy intersects terrain features as depicted in Figure E.14. The result is an increase in radar reflectivity values that can result in higher than normal precipitation estimates. The WDT processing algorithms account for these issues, but SPAS uses GIS spatial interpolation functions to infill areas suffering from poor or no radar coverage.
- **Anomalous Propagation (AP):** AP is false reflectivity echoes produced by unusual rates of refraction in the atmosphere. WDT algorithms remove most of the AP and false echoes, however in extreme cases the air near the ground may be so cold and dense that a radar beam that starts out moving upward is bent all the way down to the ground. This produces erroneously strong echoes at large distances from the radar. Again, equipped with sufficient gauge data, the SPAS bias corrections will overcome AP issues.

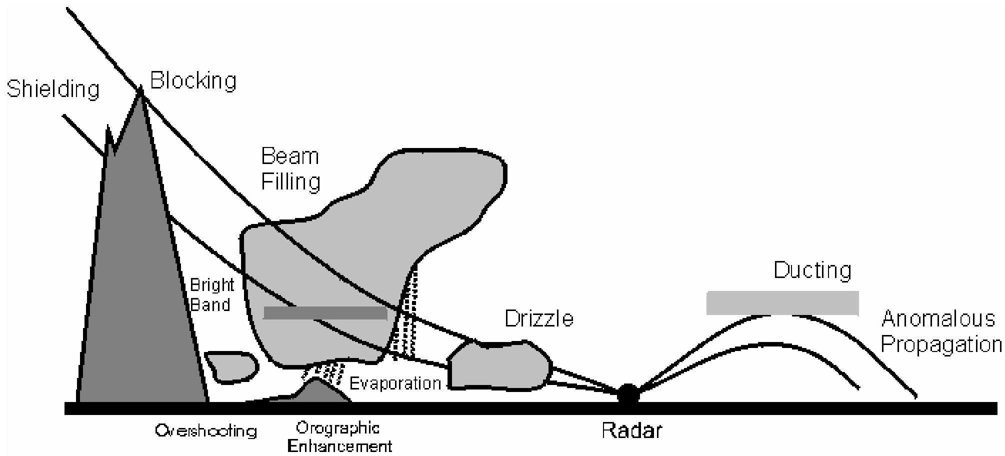


Figure E.14: Depiction of radar artifacts. (Source: Wikipedia)

SPAS is designed to overcome many of these short-comings by carefully using radar data for defining the spatial patterns and relative magnitudes of precipitation, but allowing measured precipitation values ("ground truth") at gauges to govern the magnitude. When absolutely necessary, the observed precipitation values at gauges are nudged up (or down) to force SPAS results to be consistent with observed gauge values. Nudging gauge precipitation values helps to promote better consistency between the gauge value and the grid-cell value, even though these two values sometimes should not be the same since they are sampling different area sizes. For reasons discussed in the "SPAS versus Gauge Precipitation" section, the gauge value and grid-cell value can vary. Plus, SPAS is designed to toss observed individual hourly values that are grossly inconsistent with radar data, hence driving a difference between the gauge and grid-cell. In general, when the gauge and grid-cell value differ by more than 15% and/or 0.50 inches, and the gauge data have been validated, then it is justified to artificially increase or decrease slightly the observed gauge value to "force" SPAS to derive a grid-cell value equal to the observed value. Sometimes simply shifting the gauge location to an adjacent grid-cell resolves the problems. Regardless, a large gauge versus grid-cell difference is a "red flag" and sometimes the result of an erroneous gauge value or a mis-located gauge, but in some cases the difference can only be resolved by altering the precipitation value.

Before results are finalized, a precipitation intensity check is conducted to ensure the spatial patterns and magnitudes of the maximum storm intensities at 1-, 6-, 12-, etc. hours are consistent with surrounding gauges and published reports. Any erroneous data are corrected and SPAS re-run. Considering all of the QA/QC checks in SPAS, it typically requires 5-15 basemap SPAS runs and, if radar data are available, another 5-15 radar-aided runs, to arrive at the final output.

Test Cases

To check the accuracy of the DAD software, three test cases were evaluated.

"Pyramidville" Storm

The first test was that of a theoretical storm with a pyramid shaped isohyetal pattern. This case was called the Pyramidville storm. It contained 361 hourly stations, each occupying a single grid-cell. The configuration of the Pyramidville storm (see Figure E.15) allowed for uncomplicated and accurate calculation of the analytical DA truth independent of the DAD software. The main motivation of this case was to verify that the DAD software was properly computing the area sizes and average depths.

1. Storm center: 39°N 104°W
2. Duration: 10-hours
3. Maximum grid-cell precipitation: 1.00"
4. Grid-cell resolution: 0.06 sq.-miles (361 total cells)
5. Total storm size: 23.11 sq-miles
6. Distribution of precipitation:

   Hour 1: Storm drops 0.10" at center (area 0.06 mi^2)

   Hour 2: Storm drops 0.10" over center grid-cell AND over one cell width around hour 1 center

   Hours 3-10:

   1. Storm drops 0.10" per hour at previously wet area, plus one cell width around previously wet area
   2. Area analyzed at every 0.10"
   3. Analysis resolution: 15-sec (~.25 mi^2)

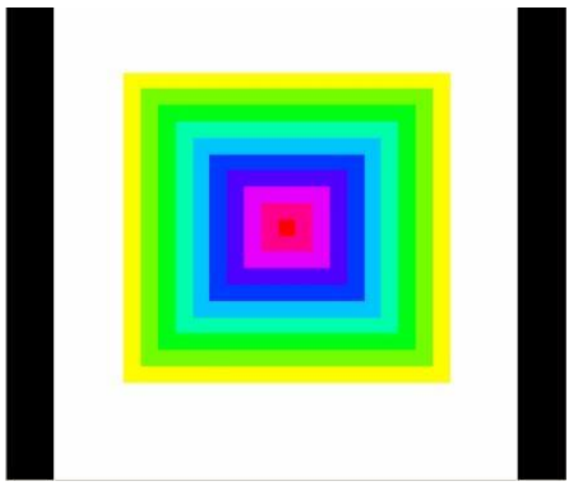


Figure E.15: "Pyramidville" Total precipitation. Center = 1.00", Outside edge = 0.10"

The analytical truth was calculated independent of the DAD software, and then compared to the DAD output. The DAD software results were equal to the truth, thus demonstrating that the DA estimates were properly calculated (Figure E.16).

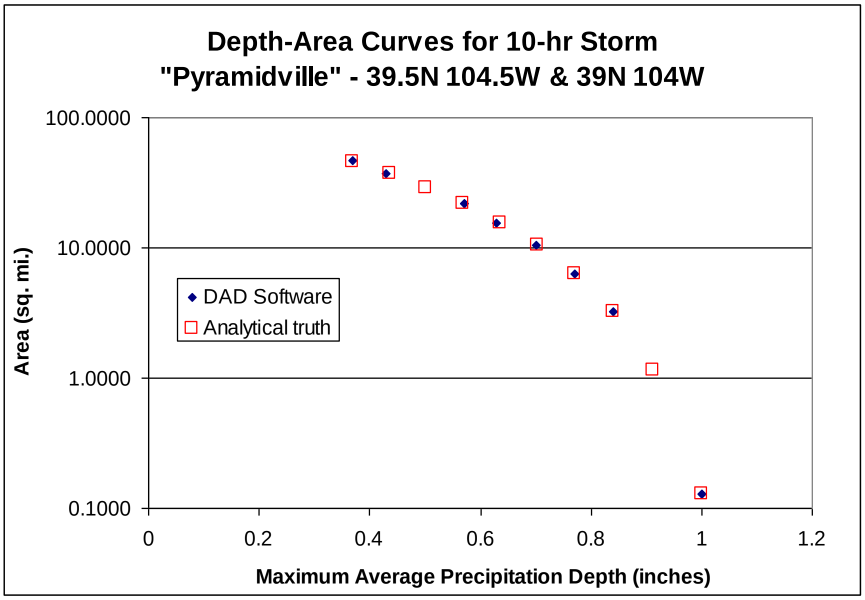


Figure E.16: 10-hour DA results for "Pyramidville"; truth vs. output from DAD software

The Pyramidville storm was then changed such that the mass curve and spatial interpolation methods would be stressed. Test cases included:

- Two-centers, each center with 361 hourly stations
- A single center with 36 hourly stations, 0 daily stations
- A single center with 3 hourly stations and 33 daily stations

As expected, results began shifting from the 'truth,' but minimally and within the expected uncertainty.

Ritter, Iowa Storm, June 7, 1953

Ritter, Iowa was chosen as a test case for a number of reasons. The NWS had completed a storm analysis, with available DAD values for comparison. The storm occurred over relatively flat terrain, so orographics were not an issue. An extensive "bucket survey" provided a great number of additional observations from this event. Of the hundreds of additional reports, about 30 of the most accurate reports were included in the DAD analysis. The DAD software results are very similar to the NWS DAD values (Table E.2).

Table E.2: The percent difference [(AWA-NWS)/NWS] between the AWA DA results and those published by the NWS for the 1953 Ritter, Iowa storm.

% Difference

		Duration (hours)			
Area (sq.mi.)		6	12	24	total
10		-15%	-7%	2%	2%
100		-7%	-6%	1%	1%
200		2%	0%	9%	9%
1000		-6%	-7%	4%	4%
5000		-13%	-8%	2%	2%
10000		-14%	-6%	0%	0%

Westfield, Massachusetts Storm, August 8, 1955

Westfield, Massachusetts was also chosen as a test case for a number of reasons. It is a probable maximum precipitation (PMP) driver for the northeastern United States. Also, the Westfield storm was analyzed by the NWS and the DAD values are available for comparison. Although this case proved to be more challenging than any of the others, the final results are very similar to those published by the NWS (Table E.3).

Table E.3: The percent difference [(AWA-NWS)/NWS] between the AWA DA results and those published by the NWS for the 1955 Westfield, Massachusetts storm

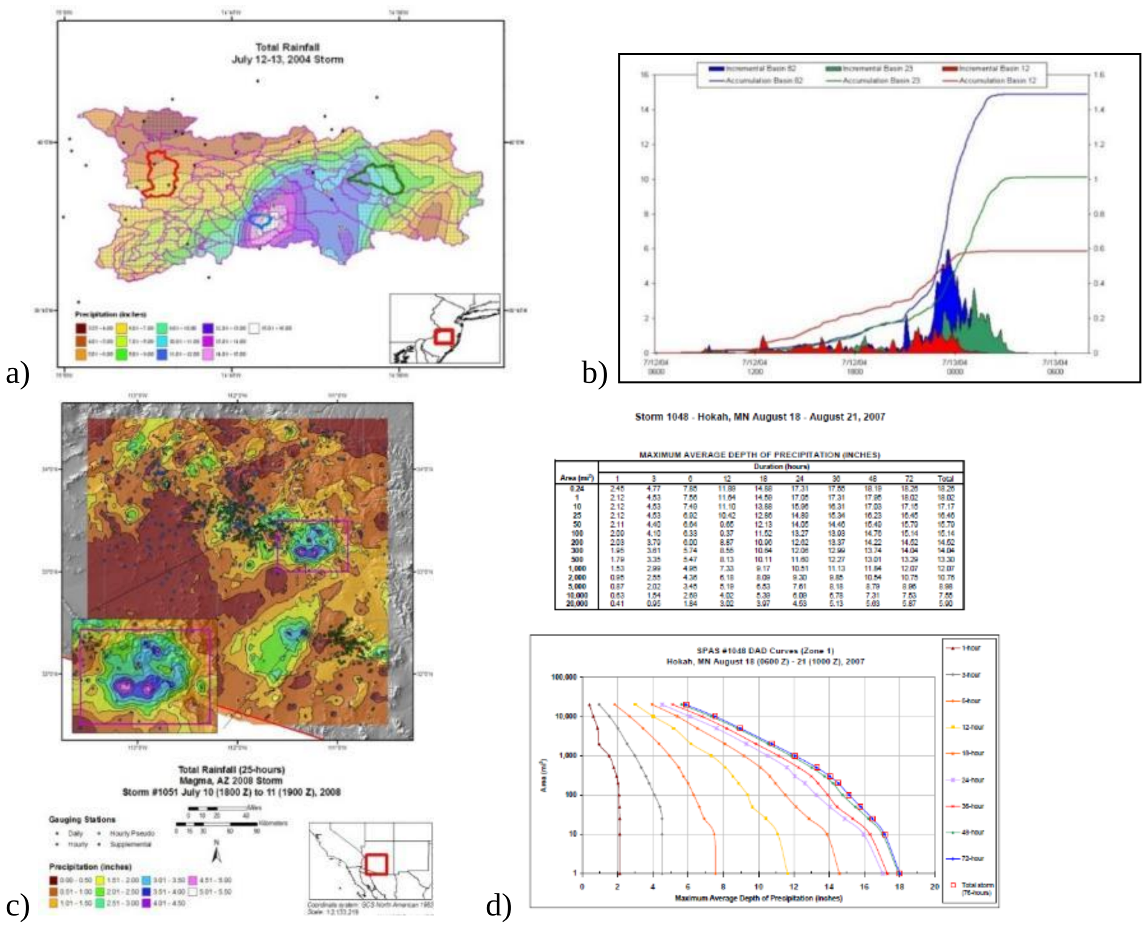
% Difference

	Duration (hours)							
Area (sq. mi.)		6	12	24	36	48	60	total
10		2%	3%	0%	1%	-1%	0%	2%
100		-5%	2%	4%	-2%	-6%	-4%	-3%
200		-6%	1%	1%	-4%	-7%	-5%	-5%
1000		-4%	-2%	1%	-6%	-7%	-6%	-3%
5000		3%	2%	-3%	-3%	-5%	-5%	0%
10000		4%	9%	-5%	-4%	-7%	-5%	1%
20000		7%	12%	-6%	-3%	-4%	-3%	3%

The primary components of SPAS are: storm search, data extraction, quality control (QC), conversion of daily precipitation data into estimated hourly data, hourly and total storm precipitation grids/maps and a complete storm-centered DAD analysis.

Output

Armed with accurate, high-resolution precipitation grids, a variety of customized output can be created (see Figures E.17A-D). Among the most useful outputs are sub-hourly precipitation grids for input into hydrologic models. Sub-hourly (i.e., 5-minute) precipitation grids are created by applying the appropriate optimized hourly Z-R (scaled down to be applicable for instantaneous Z) to each of the individual 5-minute radar scans; 5-minutes is often the native scan rate of the radar in the US. Once the scaled Z-R is applied to each radar scan, the resulting precipitation is summed up. The proportion of each 5-minute precipitation to the total 1-hour radar-aided precipitation is calculated. Each 5-minute proportion (%) is then applied to the quality controlled, bias corrected 1-hour total precipitation (created above) to arrive at the final 5-minute precipitation for each scan. This technique ensures the sum of 5-minute precipitation equals that of the quality controlled, bias corrected 1-hour total precipitation derived initially. Depth-area-duration (DAD) tables/plots, shown in Figure E.17d, are computed using a highly computational extension to SPAS. DADs provide an objective three-dimensional (magnitude, area size, and duration) perspective of a storms' precipitation. SPAS DADs are computed using the procedures outlined by the NWS Technical Paper 1 (1946).



MAXIMUM AVERAGE DEPTH OF PRECIPITATION (INCHES)

Area (mi^2)	Duration (hours)									
	1	3	6	12	18	24	36	48	72	Total
0.24	2.45	4.77	7.85	11.88	14.68	17.31	17.55	18.19	18.26	18.26
1	2.12	4.53	7.56	11.64	14.59	17.05	17.31	17.95	18.02	18.02
10	2.12	4.53	7.40	11.10	13.88	15.95	16.31	17.03	17.15	17.17
25	2.12	4.53	6.92	10.42	12.86	14.80	15.34	16.23	16.45	16.46
50	2.11	4.40	6.64	9.65	12.13	14.05	14.46	15.49	15.70	15.70
100	2.00	4.10	6.33	9.37	11.52	13.27	13.93	14.76	15.14	15.14
200	2.03	3.79	6.00	8.87	10.96	12.62	13.37	14.22	14.52	14.52
300	1.95	3.61	5.74	8.55	10.64	12.06	12.99	13.74	14.04	14.04
500	1.79	3.35	5.47	8.13	10.11	11.60	12.27	13.01	13.29	13.30
1,000	1.53	2.99	4.95	7.33	9.17	10.51	11.13	11.84	12.07	12.07
2,000	0.95	2.55	4.36	6.18	8.09	9.30	9.85	10.54	10.75	10.76
5,000	0.87	2.02	3.45	5.19	6.53	7.61	8.18	8.79	8.96	8.98
10,000	0.63	1.54	2.69	4.02	5.39	6.09	6.78	7.31	7.53	7.55
20,000	0.41	0.95	1.84	3.02	3.97	4.53	5.13	5.63	5.87	5.90

Figure E.17: Various examples of SPAS output, including (a) total storm map and its associated (b) basin average precipitation time series, (c) total storm precipitation map, (d) depth-area-duration (DAD) table and plot

Summary

Grounded on years of scientific research with a demonstrated reliability in post-storm analyses, SPAS is a hydro-meteorological tool that provides accurate precipitation analyses for a variety of applications. SPAS has the ability to compute precise and accurate results by using sophisticated timing algorithms, basemaps, a variety of precipitation data and most importantly NEXRAD weather radar data (if available). The approach taken by SPAS relies on hourly, daily and supplemental precipitation gauge observations to provide quantification of the precipitation amounts while relying on basemaps and NEXRAD data (if available) to provide the spatial distribution of precipitation between precipitation gauge sites. By determining the most appropriate coefficients for the Z-R equation on an hourly basis, the approach anchors the precipitation amounts to accepted precipitation gauge data while using the NEXRAD data to distribute precipitation between precipitation gauges for each hour of the storm. Hourly Z-R coefficient computations address changes in the cloud microphysics and storm characteristics as the storm evolves. Areas suffering from limited or no radar coverage are estimated using the spatial patterns and magnitudes of the independently created basemap precipitation grids. Although largely automated, SPAS is flexible enough to allow hydro-meteorologists to make important adjustments and adapt to any storm situation.

References

Baeck M.L., Smith J.A., 1998: “Precipitation Estimation by the WSR-88D for Heavy Precipitation Events”, *Weather and Forecasting*: Vol. 13, No. 2, pp. 416–436.

Ciach, G.J., 2003: Local Random Errors in Tipping-Bucket Rain Gauge Measurements. *J. Atmos. Oceanic Technol.*, **20**, 752–759.

Corps of Engineers, U.S. Army, 1945-1973: Storm Rainfall in the United States, Depth-Area-Duration Data. Office of Chief of Engineers, Washington, D.C.

Corrigan, P., Fenn, D.D., Kluck, D.R., and J.L. Vogel, 1999: Probable Maximum Precipitation Estimates for California. *Hydrometeorological Report No. 59*, U.S. National Weather Service, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, Silver Spring, MD, 392 pp.

Dickens, J., 2003: “On the Retrieval of Drop Size Distribution by Vertically Pointing Radar”, American Meteorological Society 32nd Radar Meteorology Conference, Albuquerque, NM, October 2005.

Duchon, C.E., and G.R. Essenberg, 2001: Comparative Precipitation Observations from Pit and Above Ground Rain Gauges with and without Wind Shields, *Water Resources Research*, Vol. 37, N. 12, 3253-3263.

Faulkner, E., T. Hampton, R.M. Rudolph, and Tomlinson, E.M., 2004: Technological Updates for PMP and PMF – Can They Provide Value for Dam Safety Improvements? Association of State Dam Safety Officials Annual Conference, Phoenix, Arizona, September 26-30, 2004.

Guo, J. C. Y., Urbonas, B., and Stewart, K., 2001: Rain Catch under Wind and Vegetal Effects. ASCE, *Journal of Hydrologic Engineering*, Vol. 6, No. 1.

Hansen, E.M., Fenn, D.D., Schreiner, L.C., Stodt, R.W., and J.F., Miller, 1988: Probable Maximum Precipitation Estimates, United States between the Continental Divide and the 103rd Meridian, *Hydrometeorological Report Number 55A*, National weather Service, National Oceanic and Atmospheric Association, U.S. Dept of Commerce, Silver Spring, MD, 242 pp.

Hunter, R.D. and R.K. Meentemeyer, 2005: Climatologically Aided Mapping of Daily Precipitation and Temperature, *Journal of Applied Meteorology,* October 2005, Vol. 44, pp. 1501-1510.

Hunter, S.M., 1999: Determining WSR-88D Precipitation Algorithm Performance Using the Stage III Precipitation Processing System, Next Generation Weather Radar Program, WSR-88D Operational Support Facility, Norman, OK.

Lakshmanan, V. and M. Valente, 2004: Quality control of radar reflectivity data using satellite data and surface observations, 20th Int’l Conf. on Inter. Inf. Proc. Sys. (IIPS) for Meteor., Ocean., and Hydr., Amer. Meteor. Soc., Seattle, CD-ROM, 12.2.

Martner, B.E, and V. Dubovskiy, 2005: Z-R Relations from Raindrop Disdrometers: Sensitivity To Regression Methods And DSD Data Refinements, 32nd Radar Meteorology Conference, Albuquerque, NM, October, 2005

Tokay, A., P.G. Bashor, and V.L. McDowell, 2010: Comparison of Rain Gauge Measurements in the Mid-Atlantic Region. *J. Hydrometeor.*, **11**, 553-565.

Tomlinson, E.M., W.D. Kappel, T.W. Parzybok, B. Rappolt, 2006: Use of NEXRAD Weather Radar Data with the Storm Precipitation Analysis System (SPAS) to Provide High Spatial

Resolution Hourly Precipitation Analyses for Runoff Model Calibration and Validation, ASDSO Annual Conference, Boston, MA.

Tomlinson, E.M., and T.W. Parzybok, 2004: Storm Precipitation Analysis System (SPAS), proceedings of Association of Dam Safety Officials Annual Conference, Technical Session II, Phoenix, Arizona.

Tomlinson, E.M., R.A. Williams, and T.W. Parzybok, September 2003: Site-Specific Probable Maximum Precipitation (PMP) Study for the Great Sacandaga Lake / Stewarts Bridge Drainage Basin, Prepared for Reliant Energy Corporation, Liverpool, New York.

Tomlinson, E.M., R.A. Williams, and T.W. Parzybok, September 2003: Site-Specific Probable Maximum Precipitation (PMP) Study for the Cherry Creek Drainage Basin, Prepared for the Colorado Water Conservation Board, Denver, CO.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D., Muhlestein, G., and B. Rappolt, May 2008: Site-Specific Probable Maximum Precipitation (PMP) Study for the Wanahoo Drainage Basin, Prepared for Olsson Associates, Omaha, Nebraska.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D., Muhlestein, G., and B. Rappolt, June 2008: Site-Specific Probable Maximum Precipitation (PMP) Study for the Blenheim Gilboa Drainage Basin, Prepared for New York Power Authority, White Plains, NY.

Tomlinson, E.M., Kappel W.D., and T.W. Parzybok, February 2008: Site-Specific Probable Maximum Precipitation (PMP) Study for the Magma FRS Drainage Basin, Prepared for AMEC, Tucson, Arizona.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D., Muhlestein, G., and P. Sutter, December 2008: Statewide Probable Maximum Precipitation (PMP) Study for the state of Nebraska, Prepared for Nebraska Dam Safety, Omaha, Nebraska.

Tomlinson, E.M., Kappel, W.D., and Tye W. Parzybok, July 2009: Site-Specific Probable Maximum Precipitation (PMP) Study for the Scoggins Dam Drainage Basin, Oregon.

Tomlinson, E.M., Kappel, W.D., and Tye W. Parzybok, February 2009: Site-Specific Probable Maximum Precipitation (PMP) Study for the Tuxedo Lake Drainage Basin, New York.

Tomlinson, E.M., Kappel, W.D., and Tye W. Parzybok, February 2010: Site-Specific Probable Maximum Precipitation (PMP) Study for the Magma FRS Drainage Basin, Arizona.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D.M., Muhlestein, G.A., March 2011: Site-Specific Probable Maximum Precipitation Study for the Tarrant Regional Water District, Prepared for Tarrant Regional Water District, Fort Worth, Texas.

Tomlinson, E.M., Kappel, W.D., Hultstrand, D.M., Muhlestein, G.A., and T. W. Parzybok, November 2011: Site-Specific Probable Maximum Precipitation (PMP) Study for the Lewis River basin, Washington State.

Tomlinson, E.M., Kappel, W.D., Hultstrand, D.M., Muhlestein, G.A., and T. W. Parzybok, December 2011: Site-Specific Probable Maximum Precipitation (PMP) Study for the Brassua Dam basin, Maine.

U.S. Weather Bureau, 1946: Manual for Depth-Area-Duration analysis of storm precipitation. *Cooperative Studies Technical Paper No. 1*, U.S. Department of Commerce, Weather Bureau, Washington, D.C., 73pp

Appendix F
Storm Data (Separate Binding)

Appendix G
GIS PMP Tool Documentation

1. PMP Tools Description and Usage

The PMP Evaluation Tools employed in this study are based on a Python script designed to run within the ArcGIS environment. ESRI's ArcGIS Desktop software is required to run the tool. The tool is compatible with ArcMap, ArcCatalog, or ArcGIS Pro. It is recommended that the most current version of the software is used. The PMP, Spatial Distribution, and Snowmelt tools provide gridded output at a spatial resolution of 90 arc-seconds (equivalent to .025 x .025 decimal degrees) for a user-designated basin or area at user-specified durations. Standard outputs include gridded and basin average PMP depths and temporally distributed accumulations. ESRI's Spatial Analyst extension is required for the Spatial Distribution and Snowmelt tools.

1.1 File Structure

The PMP tool, source script, and the storm databases are stored within the 'PMP_Evaluation_Tool' project folder. The file and directory structure within the 'PMP_Evaluation_Tool' folder should be maintained as provided, as the script will locate various data based on its relative location within the project folder. If the subfolders or geodatabases within are relocated or renamed, then the script must be updated to account for these changes.

The file structure consists of three subfolders: Input, Output, and Script. The 'Input' folder contains all input GIS files (Figure 1.1). There are six ArcGIS file geodatabase containers within the 'Input' folder: DAD_Tables.gdb, Non_Storm_Data.gdb, Spatial_Distribution.gdb, Storm_Adj_Factors.gdb, SWE.gdb, and Temperature.gdb. The DAD_Tables.gdb contains the DAD tables (in file geodatabase table format) for each of the SPAS-analyzed storm DAD zones included in the storm database. The Storm_Adj_Factors.gdb contains a feature class for each storm center and stores the adjustment factors for each grid point as a separate feature. These feature classes are organized into feature datasets, according to storm type (General, Local, and Cool-Season). The storm adjustment factor feature classes share their name with their DAD Table counterpart. The naming convention is SPAS_XXXX_Y, where XXXX is the SPAS storm ID number and Y is the DAD zone number. In the case of a hybrid storm (i.e., a storm that is run as both a general and local storm type), there will be a suffix "_gen" or "_loc" to differentiate the storm type specific to the adjustment factors in the feature class. The Non_Storm_Data.gdb contains spatial data not directly relating to the input rainfall depth or adjustment factors such as the grid network vector files. The Spatial_Distribution.gdb contains the total storm rainfall raster files for storms used by the spatial distribution tool. The SWE.gdb contains the gridded 100-year snow water equivalent (SWE) datasets used by the snowmelt runoff tool. Finally, the Temperature.gdb contains the gridded average daily temperature datasets used by the snowmelt runoff tool.

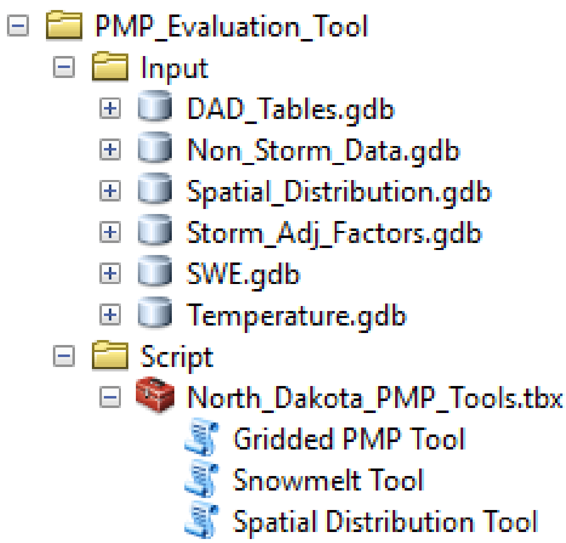


Figure 1.1: PMP tool file structure

The ‘Script’ folder contains an ArcToolbox called North_Dakota_PMP_Tools.tbx. The toolbox contains a script tool called ‘Gridded PMP Tool’ that is used to calculate PMP, a script called ‘Spatial Distribution Tool’, and a script called ‘Snowmelt Tool’. The PMP Tool will calculate gridded all-season and cool-season PMP depths in inches for a basin or user specified area size. The Spatial Distribution Tool will spatially redistribute the gridded PMP based on actual storm patterns when required. The Snowmelt Tool can be run to calculate a gridded snowmelt time series for a basin or user specified area size. The snowmelt runoff amounts can then be added to the cool-season PMP depths to determine a total combined depth of cool-season PMP and snowmelt.

ArcGIS should be used for viewing the GIS tools file structure and interacting with the input and output geospatial data. A typical operating system’s file browser does not allow access to the geodatabase containers and cannot be used to directly run the tool.

The tools are stored within the North_Dakota_PMP_Tools.tbx. ArcToolbox opens and runs the scripts within the ArcGIS environment and can be run from ArcCatalog, ArcMap, or ArcPro map session. In addition to running as a standalone tool, the tool can be incorporated into Model Builder or be called as a sub-function of another script.

To run the tools, the user navigates to the North_Dakota_PMP_Tools.tbx toolbox, expands it, and opens the appropriate tool. The dialogue window opens, and the user populates input parameters and clicks the ‘OK’ button. The tool will run in the foreground and display text output in the Messages window. Processing time can vary greatly depending on area of interest (AOI) size, the number of durations selected, and computer hardware. Most basins generally take 10 to 20 minutes to analyze all three storm types on a typical computer interface. The tools produce PMP output described in Section 1.5.

1.2 PMP Tool Usage

The tool requires several parameters as input to define the area and durations to be analyzed. The first parameter required by the tool dialogue is a feature layer, such as a basin shapefile or feature class, designed to outline the AOI for the PMP or snowmelt analysis. If the AOI dataset does not have a surface projection, the tool will apply the Albers Equal Area projection for the purpose of calculating the AOI area size. If the feature layer has multiple features (or polygons), the tool will use the combined area as the analysis region. Only the selected polygons will be used if the tool is run from the ArcMap environment with selected features highlighted. If the AOI shapefile extends beyond the project analysis domain, PMP will only be calculated for grid cells inside the project domain. The AOI shapefile or feature class should not have any spaces or symbol characters in the filename.

The second parameter requires the path of the 'PMP_Evaluation_Tool' folder. The default location of the folder is set within the tool parameters, but it can be changed if the user wishes to link the tool to another set of input datasets. The 'PMP_Evaluation_Tool' project folder should be stored locally at a location that can be accessed (both read/write permissions) by ArcGIS. The user then will need to set the 'Output Folder' path which provides the tool with the location to create the output PMP files. The user must have read/write privileges for this folder location. Note, the tool will overwrite the previous output if all input parameters are the same. The user then selects the durations to be run for each storm type. Individual durations can be run by checking each individual box or all durations can be run by clicking the "Select All" option (Figure 1.2).

The next parameter allows the user to either use the basins calculated area size or override the default to enter a custom area (in square miles) for areal-average PMP calculations. The user then has the option to have the tool perform a weighted analysis on the grid cells underlying the AOI boundary. If this option is checked each grid cell along the basin's boundary will be weighted by the portion of the cell's area inside the basin for the purpose of the basin average PMP table calculations. It is checked by default. If this option is disabled, the tool will output a basin average of all grid cells equally that intersect the basin boundary. There is an option to include sub-basin averages. This will calculate an average PMP depth for each feature in the input basin feature class from the overall basin PMP. The average sub-basin depths will be based on the area-size of the overall basin. If the 'weighted' option was selected above it will also be applied to the sub-basin averages. The user must select a field within the AOI to be used to identify each sub-basin. The field can be of numeric or text data type but must have a unique ID for each polygon. This option is disabled by default. The user can also choose to include a depth-duration chart .png image in the output folder for each storm type. Finally, the user can select the option to apply the appropriate temporal distribution patterns to the basin average PMP for each storm type. This function needs all durations of PMP to be calculated, so if this option is selected the tool will automatically run all durations for all storm types regardless of what durations were selected by the user in the previous steps (Figure 1.3).

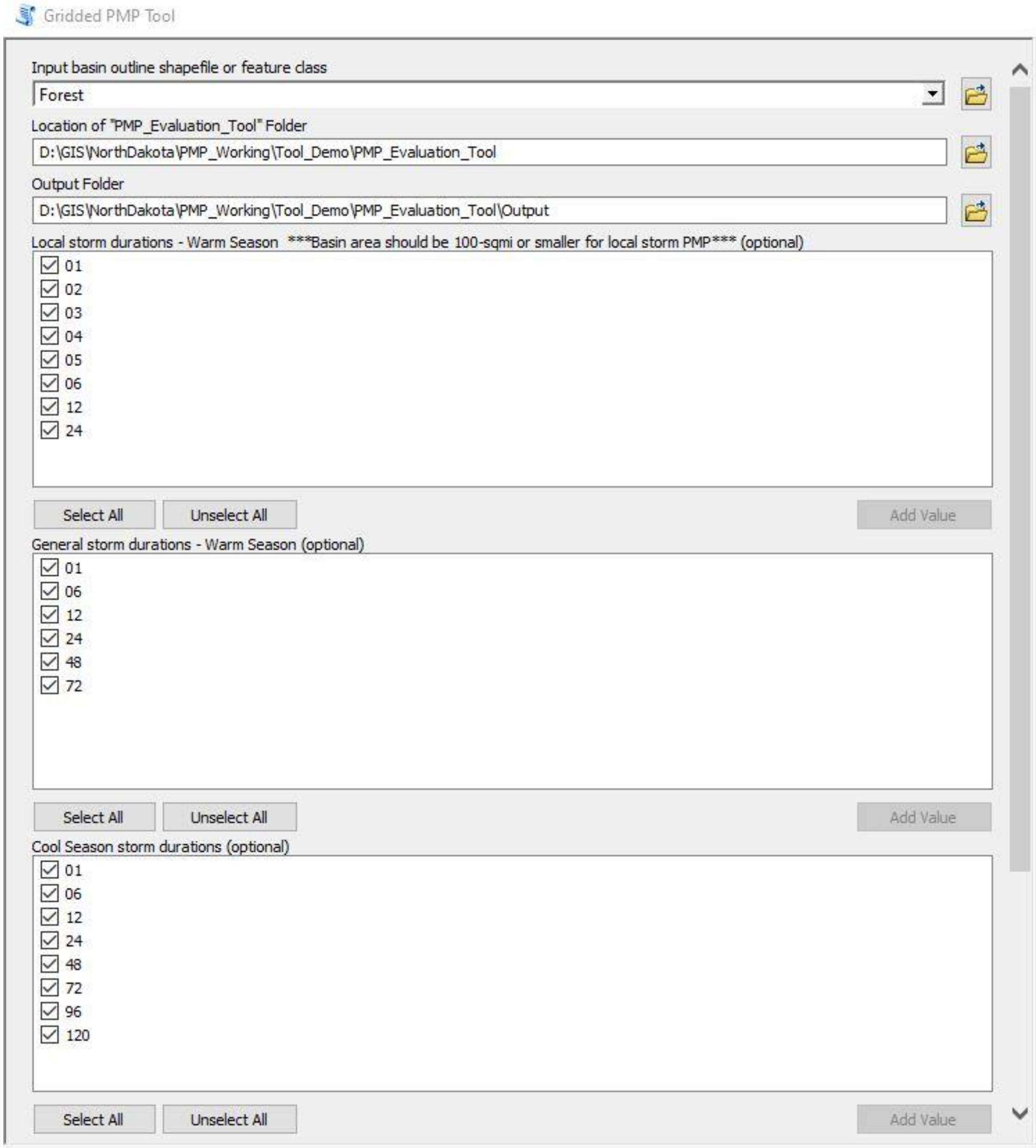


Figure 1.2: PMP tool input/output parameters with all durations set to run for the Forest drainage basin in eastern North Dakota

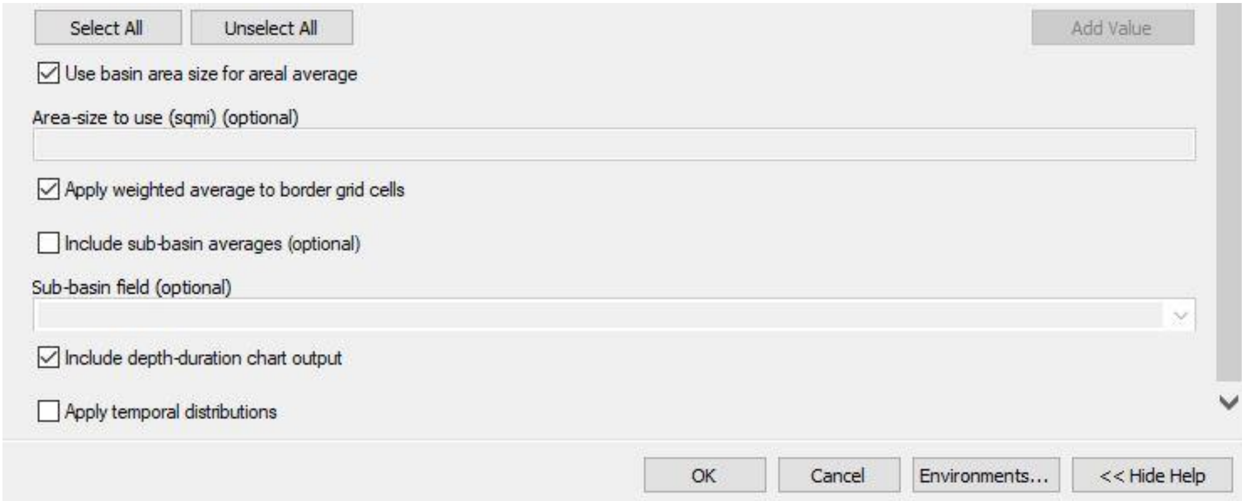


Figure 1.3: PMP tool input/output parameters with default options enabled

1.3 Spatial Distribution Tool Usage

The Spatial Distribution Tool (Figure 1.4) can be run after the Gridded PMP Tool to provide alternate spatial distribution patterns over the drainage basin. The tool uses the total storm rainfall patterns from the various historical events in the PMP tool database to redistribute gridded PMP over the basin, without changing the basin average PMP. The tool can be used for any drainage area inside the project domain; however, spatial variations and their effect on PMP is nominal for smaller area sizes and therefore alternative spatial patterns are not required for basins less than 50 mi^2. By default, the Spatial Distribution Tool "centers" the spatial pattern over the centroid of the basin. The user also has the option to center the pattern elsewhere in the basin by providing coordinates for a point inside the basin. The tool applies the following default recommended spatial patterns, each of which are representative of meteorologically possible spatial patterns observed in storms used for PMP development:

<u>Local Storm:</u>

- LS - Wooster, OH, Jul. 1969 (SPAS_1209_1)
- LS - Boyden, IA, Sep. 1926 (SPAS_1427_1)
- LS - Hayward, WI, Aug. 1941 (SPAS_1699_1)

<u>General Storm:</u>

- GS - Ida Grove, IA, Aug. 1962 (SPAS_1527_1)
- GS - Council Grove, KS, Jul. 1951 (SPAS_1583_1)

<u>Cool-Season Storm:</u>

- CS - Bellefontaine, OH, Mar. 1913 (SPAS_1698_1)
- CS - Groton, SD, May. 2007 (SPAS_1733_1)

Alternatively, the user can choose spatial patterns from the list of storms in the database. If the user chooses this option, they should have enough knowledge of the various historical events to ensure they are reasonable options for the drainage basin.

The basin input should be the same basin shapefile/feature class used in the gridded PMP tool. As with the PMP tool, the second input parameter is the location of the 'PMP Evaluation Tool' folder, which should be populated automatically. The third parameter is the 'PMP_Points' feature class, which is an output from the PMP tool. The fourth parameter is the option to choose to use the basin centroid as the target center for the spatial pattern(s). This is the default choice. If this box is unchecked, the user can then enter the target center location, in degrees longitude (X Coordinate) and degrees latitude (Y Coordinate) as the fifth parameter. The user should take care to ensure this location is within the basin. The six parameter is the option to apply the default spatial patterns (listed above). This option is recommended. The spatial patterns for the appropriate storm type, determined by the 'PMP_Points' feature class from the third parameter, will be applied. If the user chooses not to use the default spatial patterns and unchecks this box, the various spatial patterns for parameter six will become available. The user can check multiple patterns, but they should correspond with the input PMP storm type (i.e., "LS" patterns for the local PMP storm type). Finally, the user chooses the output folder location for the spatially redistributed PMP.

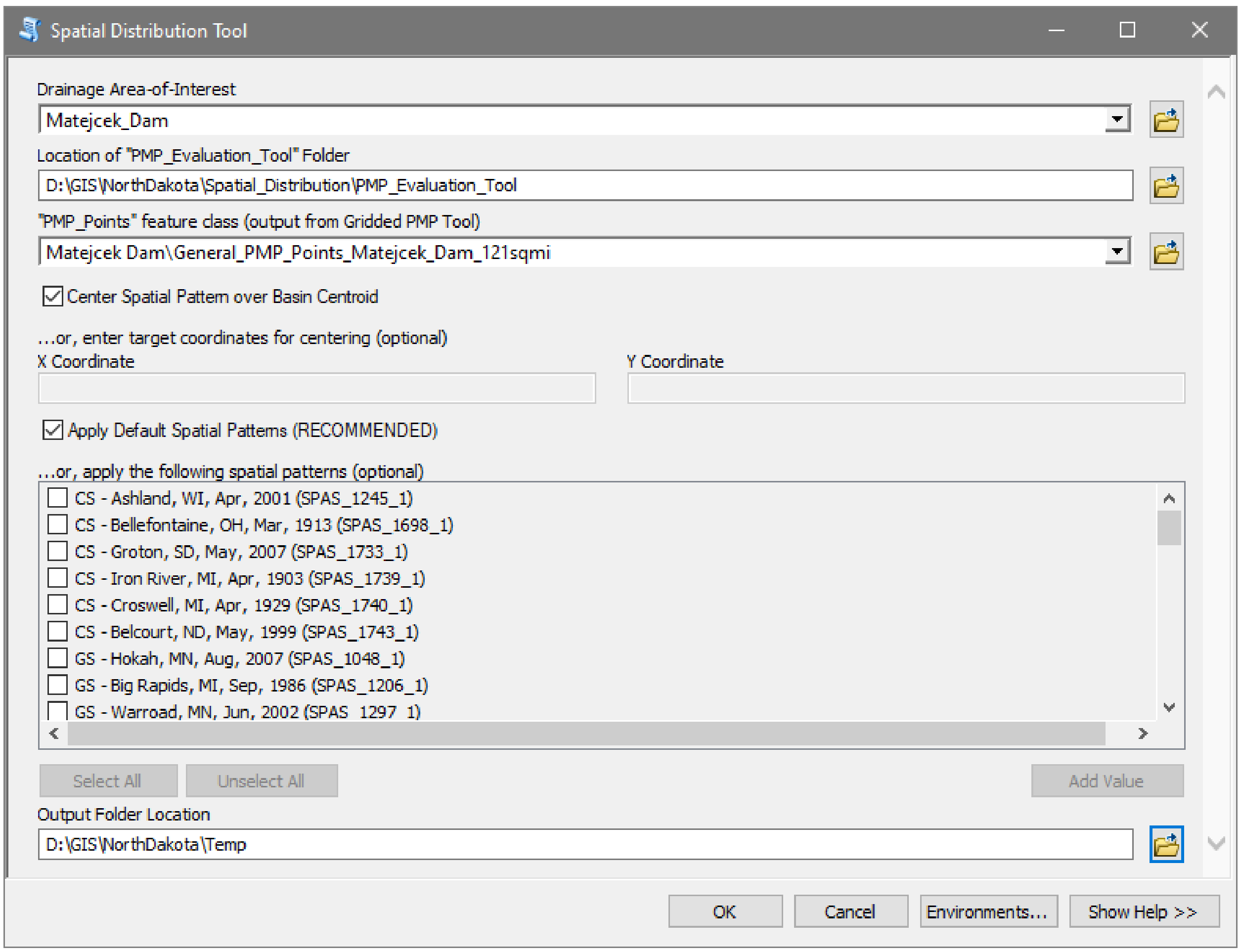


Figure 1.4: Spatial Distribution tool input/output parameters with default options enabled

1.4 Snowmelt Tool Usage

Like the PMP tool, the first parameter of the snowmelt tool also requires the path of the 'PMP_Evaluation_Tool' folder (Figure 1.5). The default location of the folder is set within the tool parameters, but it can be changed if the user wishes to link the tool to another set of input datasets. The 'PMP_Evaluation_Tool' project folder should be stored locally at a location that can be accessed (both read/write) by ArcGIS desktop. Next the user chooses a start date for the melt event. The default is March 15, but the user can type in or use the calendar to choose any date between March 1 and June 15. The start date must fall between these dates as the 100-year SWE datasets were only created for this period as significant melting is unlikely with cool-season PMP before March 1 and no snow water equivalent will be available after June 15.

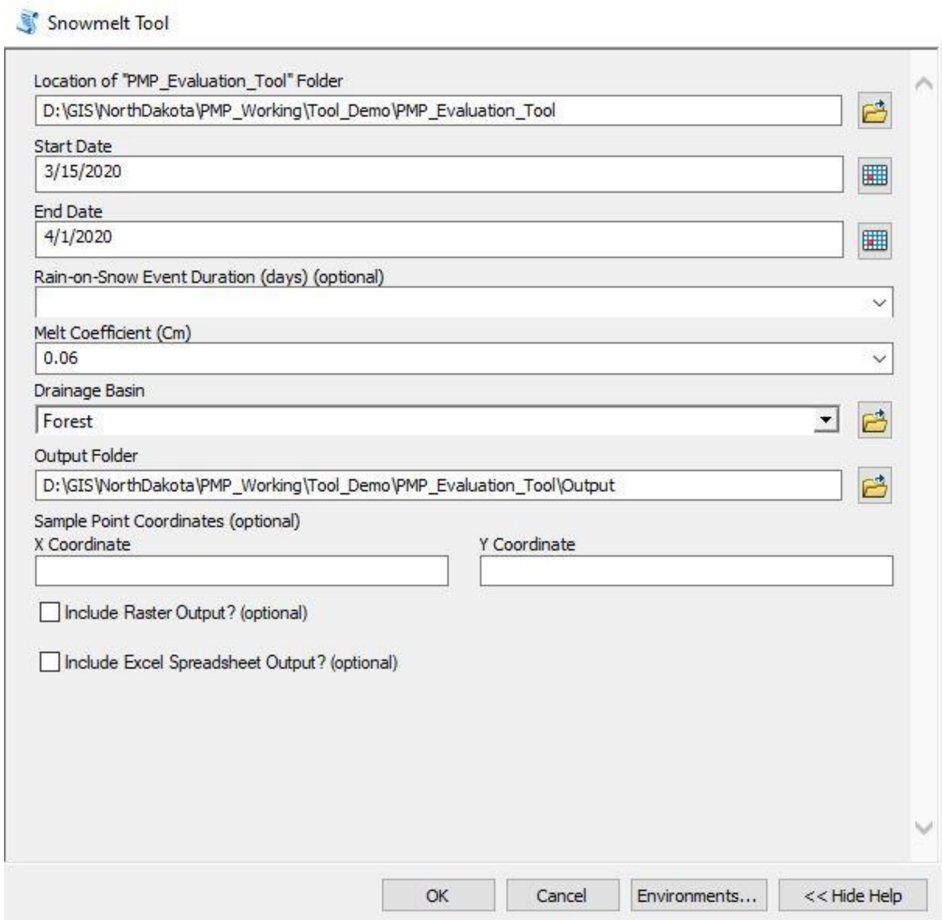


Figure 1.5: Snowmelt tool input/output parameters with default options enabled

There are multiple items to consider when choosing the start date of the melt and this will vary greatly by basin location and area size. Cool-season PMP and Snowmelt are not required for basins less than 100-mi^2. The user should apply knowledge of the critical time of the year with maximum SWE melt potential for a given basin. If this is not known, then several sensitivities should be run to determine the optimized dates of snowmelt. The next parameter allows the user to simulate a 1, 3, 5, or 7-day rain-on-snow event. This is an optional parameter and by default the tool does not consider this, it is enabled by a dropdown menu if the users choose to model a rain-on-snow scenario (Figure 1.6).

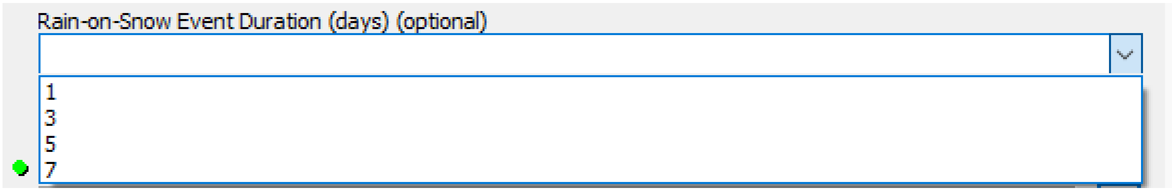


Figure 1.6: Dropdown menu for rain-on-snow option

The rain-on-snow melt function applies a maximized temperature profile to account for and represent the relatively warm and moist airmass associated with cool-season PMP rainfall events. A maximization of 5°F, based on storm maximization methods, increases the daily average temperature by 5°F for the duration of the rain-on-snow event with a 2.5°F increase the day before and the day after the event to produce a more realistic temperature sequence. For example, if the user chose a start date of March 15th with a 3-day rain-on-snow event, the first day would extract the SWE and temperature values from embedded datasets as normal. On the second day the rain-on-snow event would trigger the temperature increase by adding 2.5 °F. Then the temperatures would be increased by 5°F over the temperatures before the cool-season PMP rainfall period for the next 3 days for the rainfall event. Finally, the temperatures would drop down by 2.5°F the next day and then back to the actual extracted temperature values for the duration of the period run.

Next, the melt coefficient (C_m) sets the conditions for the melt. By default, this dropdown option uses a clear day melt coefficient. The C_m can be set to three other values (Figure 1.7) that represent different meteorologic conditions for rain-on-snow conditions.

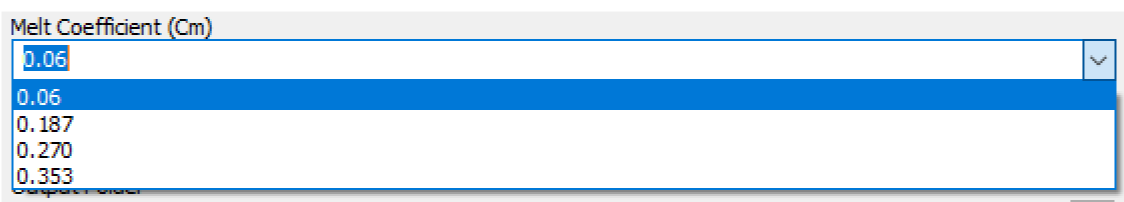


Figure 1.7: Dropdown menu for melt coefficients

There are four options to set the melt conditions using the melt coefficient.

- 0.06 – Clear sky, no rain, limited melt factor (this is the default).
- 0.187 – Heavy rain, 10 mph wind melt factor.
- 0.270 – Heavy rain, 20 mph wind melt factor.
- 0.353 – Heavy rain, 30 mph wind melt factor.

The user will then choose the input drainage basin as a polygon feature class to model, then choose the output folder location. The next optional parameter provides output for a discrete location based on the input coordinates along with the basin average values.

The standard output of the tool is an ESRI geodatabase table. The last two optional parameters will also include the output in raster and excel formats if chosen instead of just the geodatabase table of the basin average.

1.5 PMP Tool Output

Once the tool has been run, the output file geodatabases will be populated with the model results. The GIS files can then be brought into an ArcMap, or other compatible GIS environments, for mapping and analysis.

Note, the tool is set to have overwrite capabilities; if output data exists, it will be overwritten the next time the tool is run, if the same output folder and same parameters are used.

A separate output folder is created for each storm type and the output is organized within file geodatabases and named according to the input basin feature name and analyzed PMP area. Each output file geodatabase contains a feature class which stores each grid point centroid within the basin as a separate feature. Each feature has a field for the grid ID, latitude, longitude, analysis zone, elevation, PMP (for each duration), and the contributing storm ID. PMP raster files are also stored within the file geodatabase. The naming convention for the raster files is the storm type and duration (L for Local, G for General, and C for Cool-Season), followed by the input basin feature name, and ending with the basin area (in square miles). If temporal patterns were applied, the output tables will also be in the geodatabase. A folder named CSV is also created and all the geodatabase tables are exported to csv files. An example of the output file structure is shown in Figure 1.8.

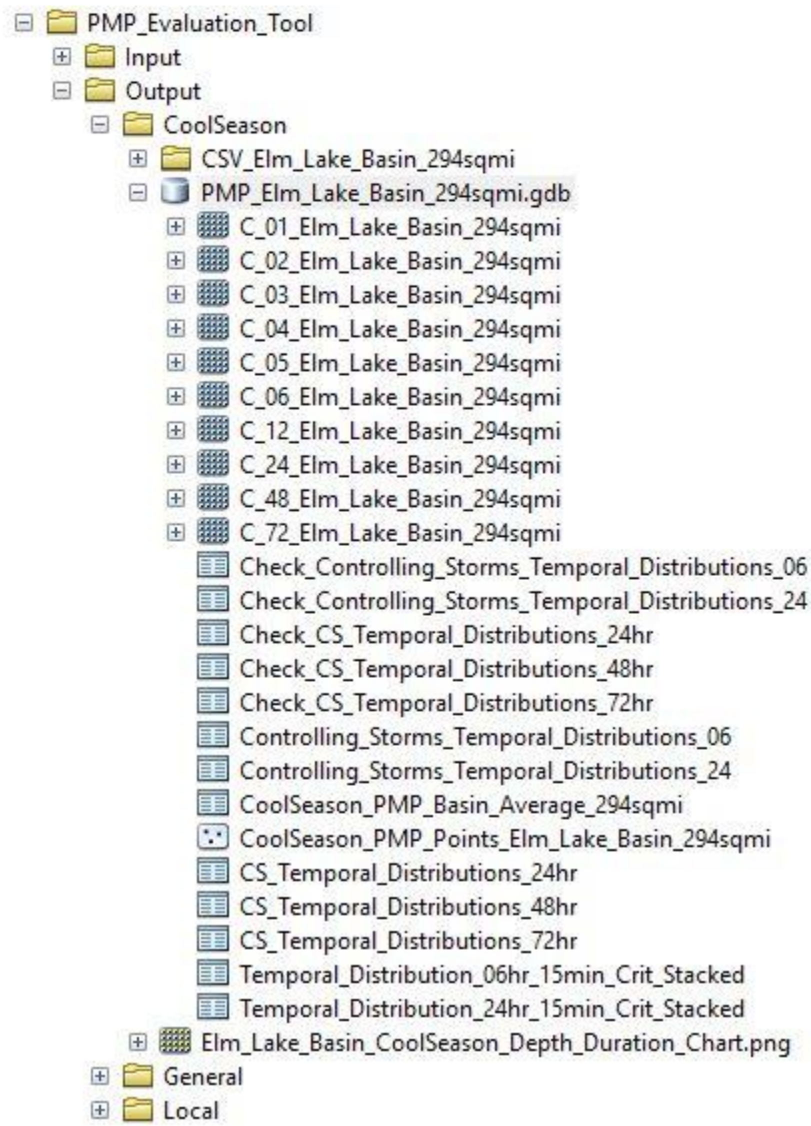
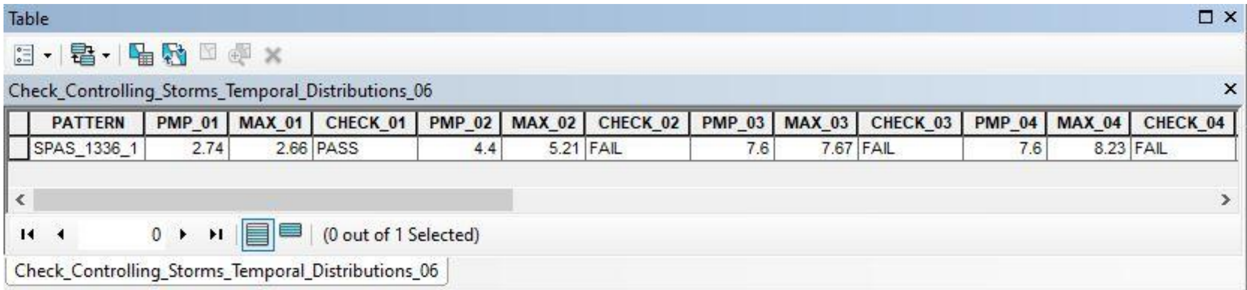


Figure 1.8: Example of the PMP tool output file structure

If the temporal patterns were applied, you will see a table named Temporal_Distribution_Check. This is important as it evaluates the temporally distributed PMP values for each duration against the PMP value for that duration. The table has a pass or fail. If the temporally distributed PMP value exceeds the PMP at a given duration, the table will have FAIL for that duration and this temporal pattern should not be applied. An example is shown in Figure 1.9.



Table

Check_Controlling_Storms_Temporal_Distributions_06

PATTERN	PMP_01	MAX_01	CHECK_01	PMP_02	MAX_02	CHECK_02	PMP_03	MAX_03	CHECK_03	PMP_04	MAX_04	CHECK_04
SPAS_1336_1	2.74	2.66	PASS	4.4	5.21	FAIL	7.6	7.67	FAIL	7.6	8.23	FAIL

(0 out of 1 Selected)

Check_Controlling_Storms_Temporal_Distributions_06

Figure 1.9: Example of the temporal check results

In the example above (Figure 1.9), the basin average 1-hour PMP is 2.74". Using the temporal distribution for one of the controlling storms, the maximum 1-hour value is 2.66". This passes the check. However, for the 2-hour PMP the maximum temporally distributed value of 5.21" is exceeding the 4.4" PMP values. This fails the check, and this pattern should not be applied to the PMP values.

1.6 Spatial Distribution Tool Output

The Spatial Distribution Tool output follows the same format as the PMP Tool output described in Section 1.5 in that there will be a "PMP_Points" feature class with the point vector PMP depths for each grid point, and gridded PMP raster files for each duration included in the original PMP output, all included within a file geodatabase. A separate file geodatabase will be created for each spatial pattern applied (either by default or chosen by the user). The naming convention is also similar but also includes the SPAS ID number to identify the spatial pattern used and the "spatial" suffix on each output file to identify as spatially redistributed. Figure 1.10 shows sample spatially distributed general storm PMP for the Matejcek Dam basin (121-square mile). There is a separate file geodatabase for both of the general storm default spatial patterns; SPAS 1527 (August 1962 event) and SPAS 1583 (July 1951 event).

Figure 1.10 – Example of Spatial Distribution Tool output

1.7 Snowmelt Tool Output

Once the tool has been run, the output file geodatabases will be populated with the model results. The GIS files can then be brought into an ArcMap, or other compatible GIS environments, for mapping and analysis. If the option to export to Excel spreadsheet was enabled, then a copy of this geodatabase will be created as an Excel file.

Note, the tool is set to have overwrite capabilities; if output data exists, it will be overwritten the next time the tool is run, if the same output folder and same parameters are used.

Based on the tools optional input parameters, the tool will create a geodatabase and populate with the tool output. The naming convention will be “Snowmelt_basin Name_Start Date”.gdb The output table will contain seven fields:

- Day – The date of the melt day
- Ta – Basin average daily temperature
- DegreeDays – Basin average daily temperature above freezing
- Cm – Melt coefficient used
- SWE – Basin average 100-year snow water equivalent based on the start date chosen
- Melt – Basin average daily melt
- MeltAccum – Basin average melt accumulation

OBJECTID *	Day	Ta	DegreeDays	Cm	SWE	Melt	MeltAccum
1	3/15/2020	24.42	0	0.06	6.06	0	0
2	3/16/2020	25.1	0	0.06	6.06	0	0
3	3/17/2020	26.21	0	0.06	6.06	0	0
4	3/18/2020	26.44	0	0.06	6.06	0	0
5	3/19/2020	27.24	0	0.06	6.06	0	0
6	3/20/2020	26.99	0	0.06	6.06	0	0
7	3/21/2020	27.87	0	0.06	6.06	0	0
8	3/22/2020	29.22	0	0.06	6.06	0	0
9	3/23/2020	29.22	0	0.06	6.06	0	0
10	3/24/2020	29.92	0	0.06	6.06	0	0
11	3/25/2020	29.66	0	0.06	6.06	0	0
12	3/26/2020	29.28	0	0.06	6.06	0	0
13	3/27/2020	30.67	0	0.06	6.06	0	0
14	3/28/2020	31.87	0.33	0.06	6.06	0.02	0.02
15	3/29/2020	32.93	1.06	0.06	6.04	0.06	0.08
16	3/30/2020	33.98	1.98	0.06	5.98	0.12	0.2
17	3/31/2020	34.28	2.29	0.06	5.86	0.14	0.34
18	4/1/2020	34.24	2.24	0.06	5.72	0.13	0.47

(0 out of 18 Selected)

Figure 1.11: Example of the Snowmelt tool output table run with default parameters for the Forest drainage basin

In the example table above the Forest basin was run with the default input parameters. The March 15th start date produced below freezing temperatures for the first thirteen days. The 6.06 inches of SWE available does not start to melt until the 14th day and only results in melting a total of 0.47 inches.

Along with the basin average output table shown in Figure 1.11, the tool will also export each field's result as a gridded geodatabase raster if chosen in input parameters.

1.8 Known Issues and Troubleshooting

The GIS PMP tool has undergone a beta testing program during development. One goal of the beta testing program was to identify possible issues with the GIS tool. The following guidelines may prevent issues with running the GIS tool.

- Ensure ArcGIS Desktop is up to date with the most recent version release and maintenance is current.
- Ensure all file and path names do not have spaces or non-alphanumeric symbols (e.g., #, $, %). Underscores are acceptable and a good alternative to using spaces.

- Close any other applications or instances of ArcMap that may interfere with the current session, files, or file paths that will be used by the tool.
- Ensure that all file paths, input and output files, and ArcGIS Environment settings (including the Default.gdb and Scratch.gdb) are local and not set to a network location.

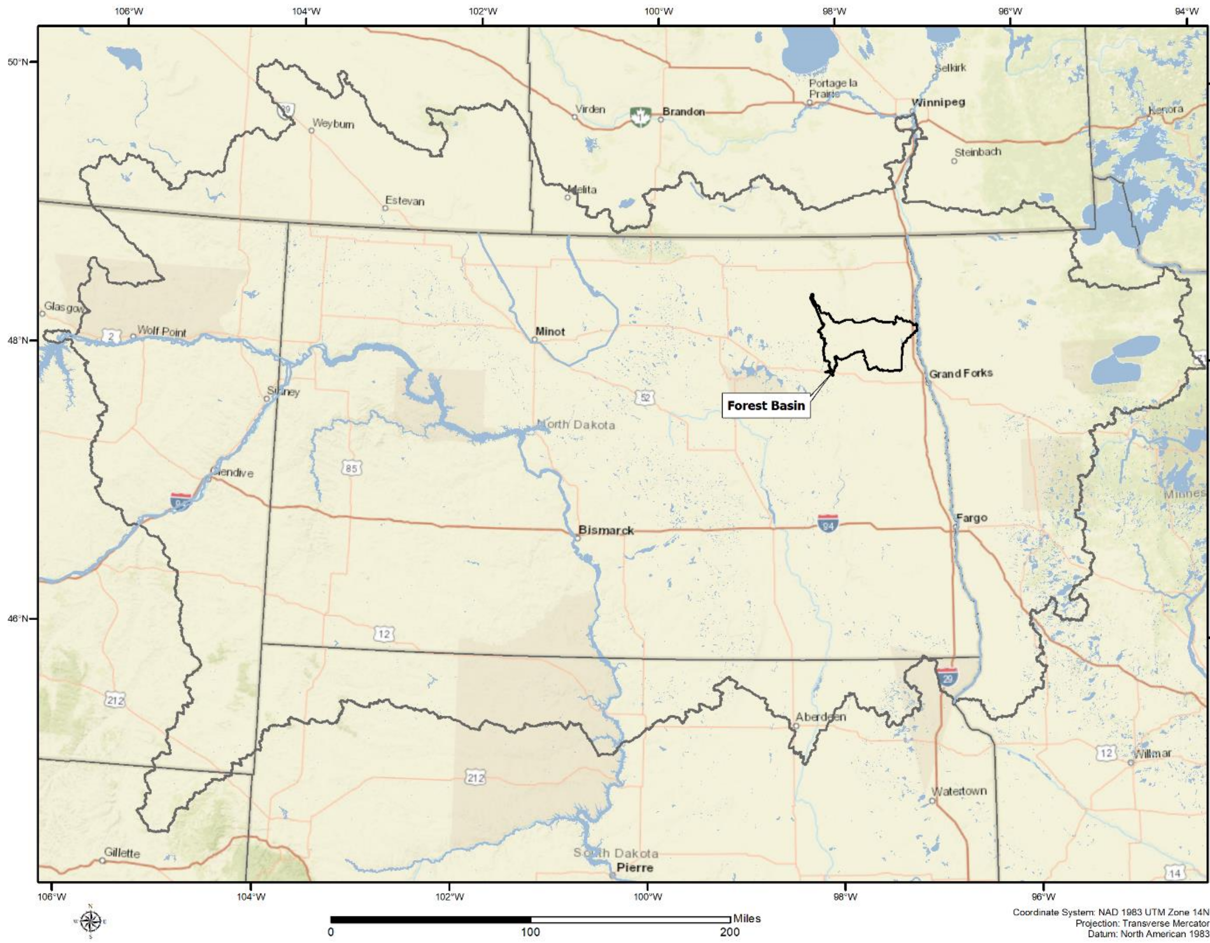
If the points above have been verified and issues persist, the user may try the following actions to address the issue:

- Close out all ArcMap sessions and all ArcGIS applications and restart session.
- Restart computer. This may be required to completely clear any locks on files or memory.
- Run the Repair Geometry tool on the AOI shapefile or feature class to correct any geometry issues within the file.
- Rename AOI file. Change tool and/or output folder paths.
- If issues persist it may be necessary to contact ESRI support or perform a clean ArcGIS installation or upgrade.

2. Sample Basin Example

2.1 PMP Tool

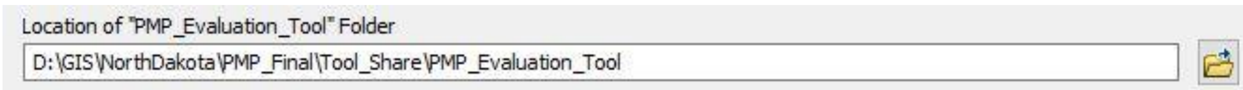
This section will walk through the steps required to run the tool for a sample basin. This example will use the Forest Basin. It is 937 square miles and is in northeastern North Dakota.



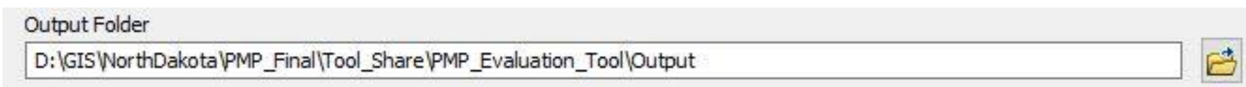
- Once downloaded add the North Dakota PMP Tool ArcToolbox to your ArcMap, ArcPro, or ArcCatalog session.

- Double click the Gridded PMP Tool script and the input dialog will appear.
- First choose the input basin. If the file is already in your project, you can choose it from the dropdown. Otherwise click on the folder to navigate to the file location.

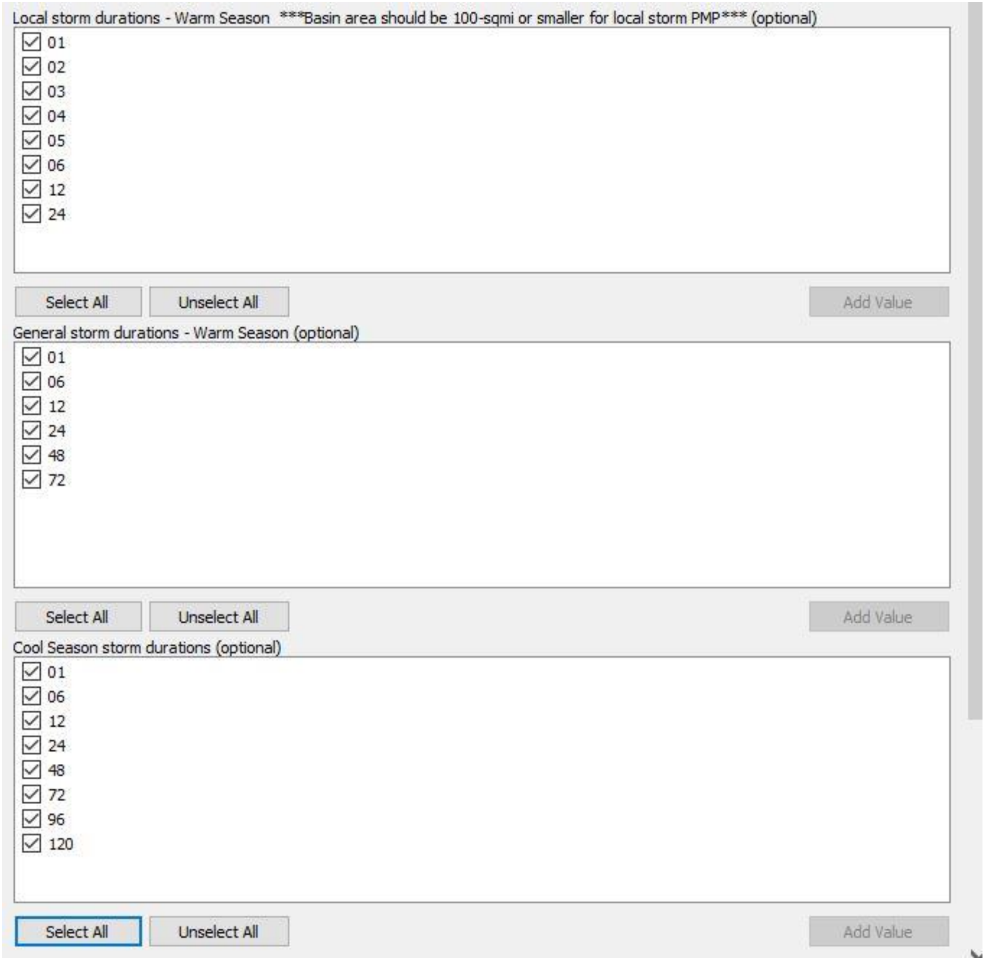
- The next parameter automatically populates with the location of the PMP_Evaluation_Tool folder.



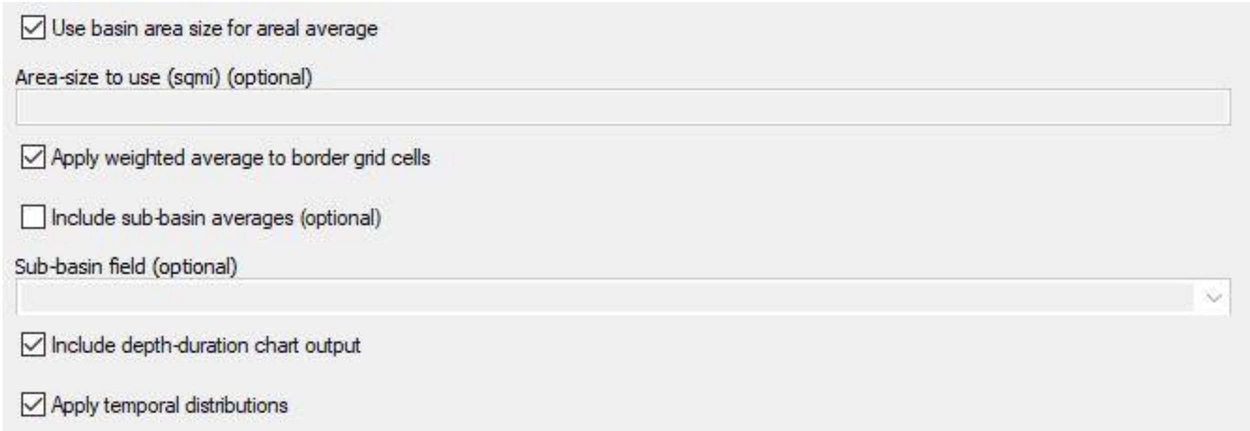
- Next navigate to a folder location to store the tool output.



- Now choose the durations to be run for each storm type. In this example we will run all durations for all storm types.



- The default options are checked for the remaining except for the option to apply temporal distributions. Check the box to apply the temporal distributions to the PMP values.

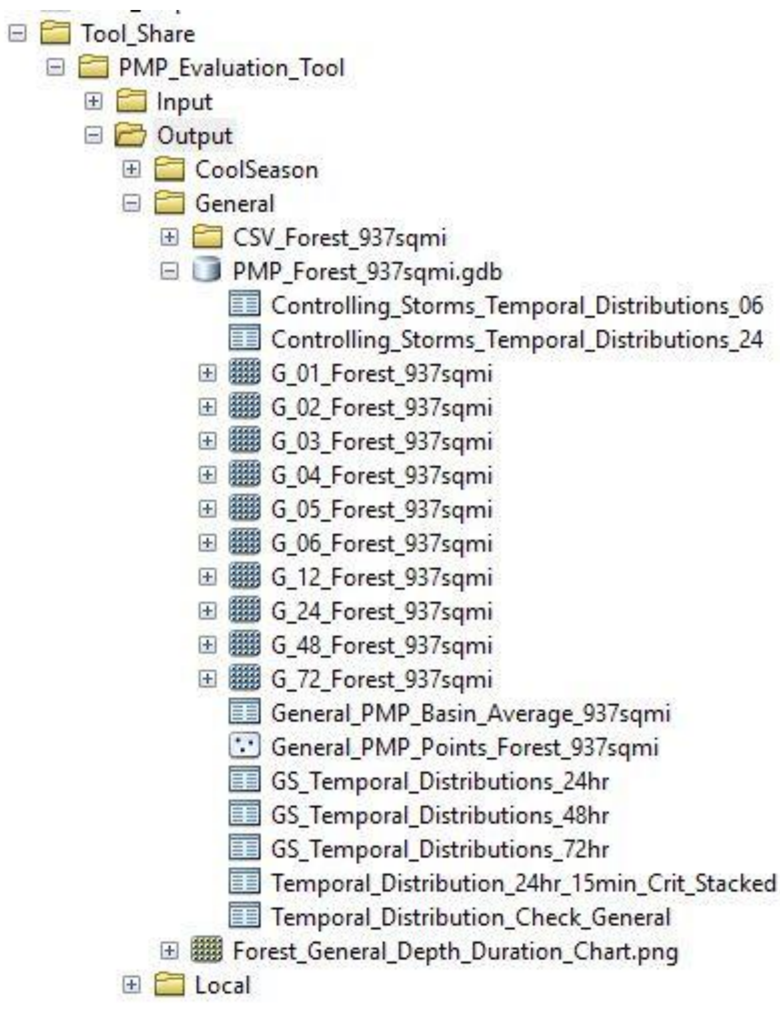


- Click ok to run the tool. The tool runs and provides feedback on the progres as the script runs. Make sure the highlighted checkbox is unchecked and you can go through the report when completed.

Close this dialog when completed successfully

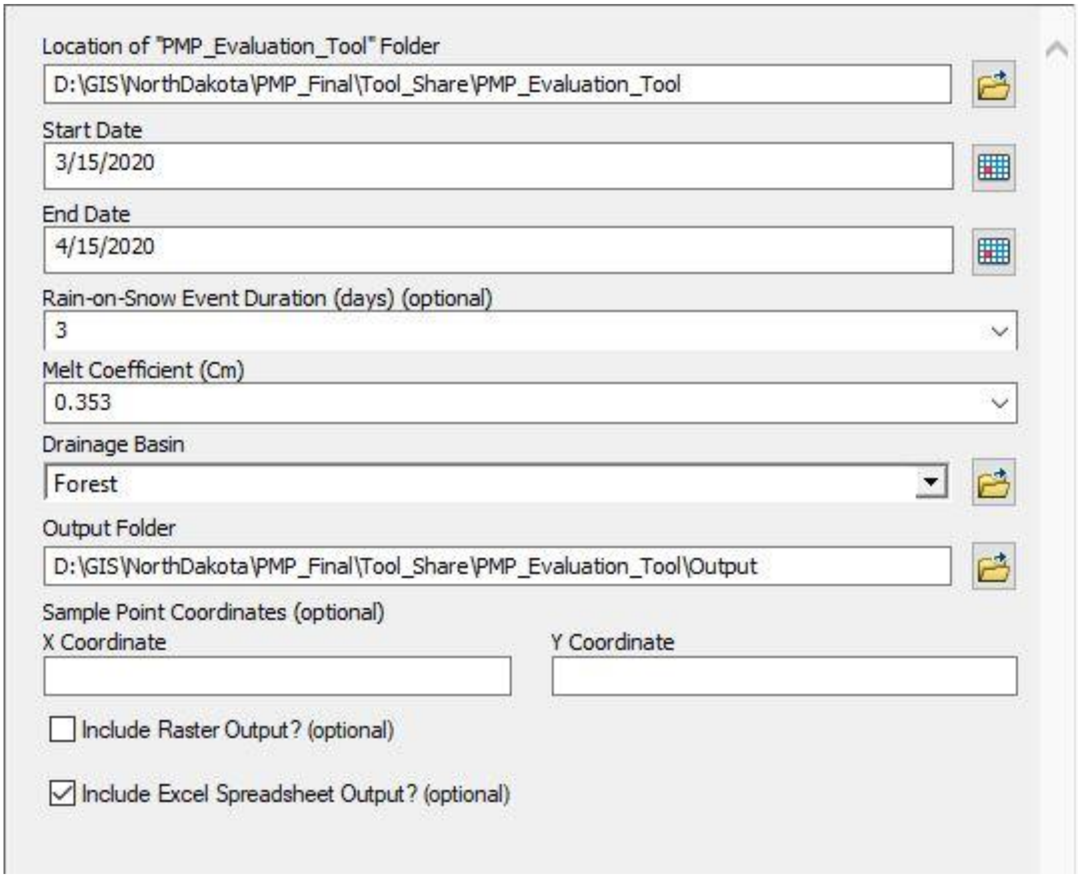
```
Running PMP analysis for storm type: Local

Basin shape type: Polygon

Creating feature class: 'PMP_Points' in Scratch.gdb...
(558 grid points will be analyzed)

     ...adding field: PMP_01
     ...adding field: PMP_02
     ...adding field: PMP_03
     ...adding field: PMP_04
     ...adding field: PMP_05
     ...adding field: PMP_06
     ...adding field: PMP_12
     ...adding field: PMP_24
     ...adding field: STORM_01
     ...adding field: STORM_02
     ...adding field: STORM_03
     ...adding field: STORM_04
     ...adding field: STORM_05
     ...adding field: STORM_06
     ...adding field: STORM_12
     ...adding field: STORM_24

AOI basin spatial reference:  NAD83 / North Dakota South (ft)
Unit type: foot
Spatial reference type: Projected

Area of interest: 936.793 square miles.

***Area used for PMP analysis: 936.8 sqmi***

***Warning - Local storm PMP depths only valid for basins 100 square
miles or smaller***

**************************************************************
Evaluating 01-hour duration...

     Evaluating storm: SPAS_1030_1...
           01-hour DAD value:  2.012"
           Comparing SPAS_1030_1 adjusted rainfall values against
current driver values...
           Transposed to 558/558 grid points...
```

- Navigate to output folder chosen in tool input dialog to explore output files.



2.2 Snowmelt Tool

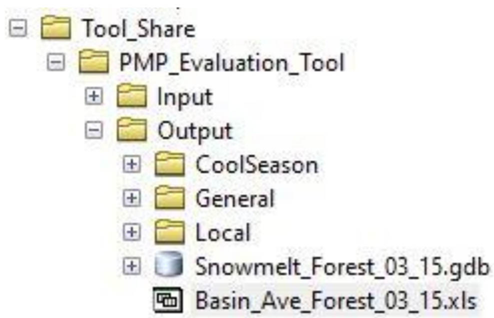
- Next run the Snowmelt Tool if needed for your specific location. Double click the Snowmelt Tool script and the tool dialog opens. In the example below the location of the PMP_Evaluation_Tool folder is already populated. Set appropriate melt start and end dates. In this example, March 15th through April 15th is utilized along with a 3-day rain-on-snow PMP event with a worst case melt coefficient of 0.353. The basin file and output location are input. We did not choose to add gridded data but instead chose to output an excel file of the basin average values.



- Click ok and the tool runs and reports values for each day in the dialog.

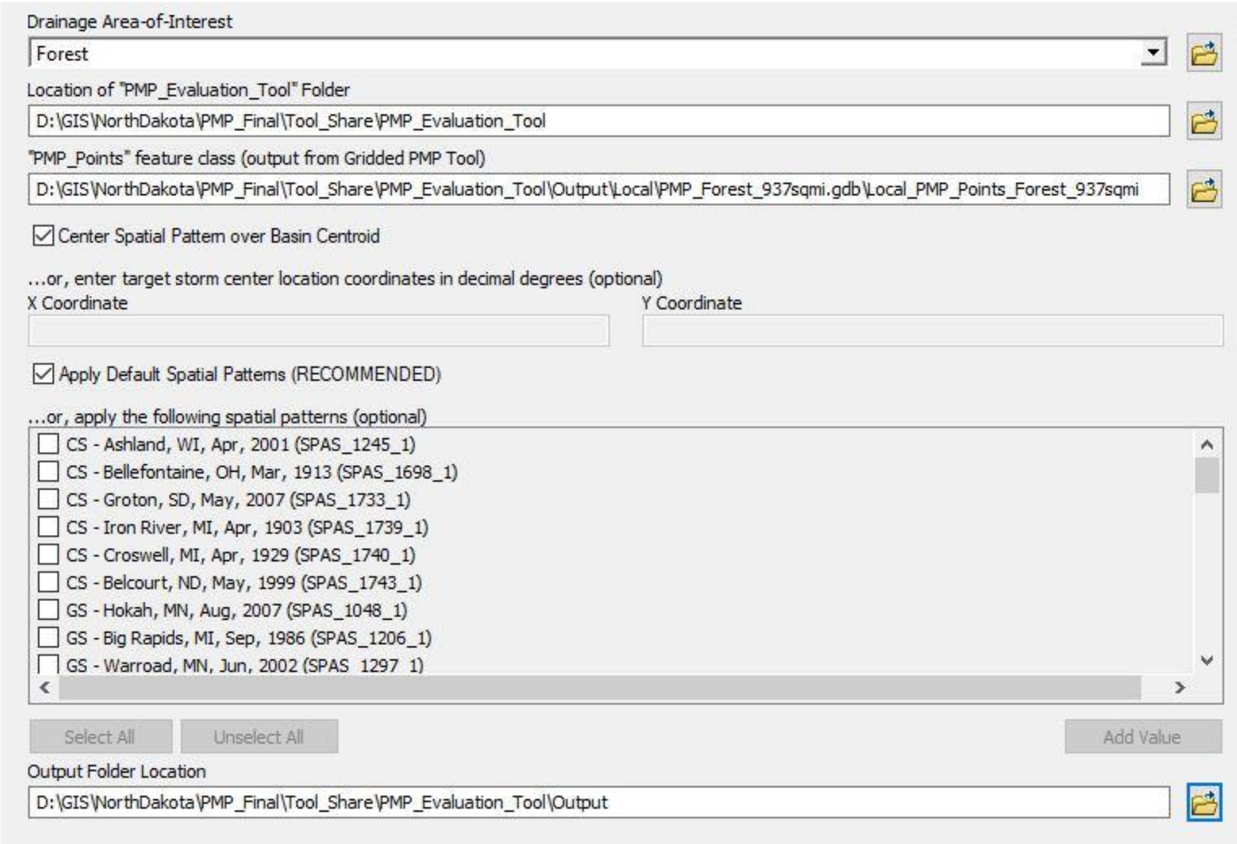
```
Melt start date: Mar, 15

Number of days: 32

Creating Output File Geodatabase: D:\GIS\NorthDakota\PMP_Final
\Tool_Share\PMP_Evaluation_Tool\Output\Snowmelt_Forest_03_15.gdb

Starting SWE Raster: ND_Final_100yr_SWE_0315

122 temperature rasters loaded.


Evaluating Day 1: Mar, 15
     Basin Average Temperature Value: 24.42
     Basin Average Degree Days Value: 0.00
     Basin Average SWE Value: 6.06
     Basin Average Melt Value: 0.00
     Basin Average Melt Accumulation Value: 0.00


Evaluating Day 2: Mar, 16

...applying 50% rainfall adjustment (1.075)
     Basin Average Temperature Value: 26.98
     Basin Average Degree Days Value: 0.00
     Basin Average SWE Value: 6.06
     Basin Average Melt Value: 0.00
     Basin Average Melt Accumulation Value: 0.00
```

- Navigate to the output folder location and we can see that the tool added a geodatabase and an excel table to the output folder. These resulting daily melt accumulations can then be added to the cool-season PMP depths to get a total amount of potential runoff.



2.3 Spatial Distribution Tool

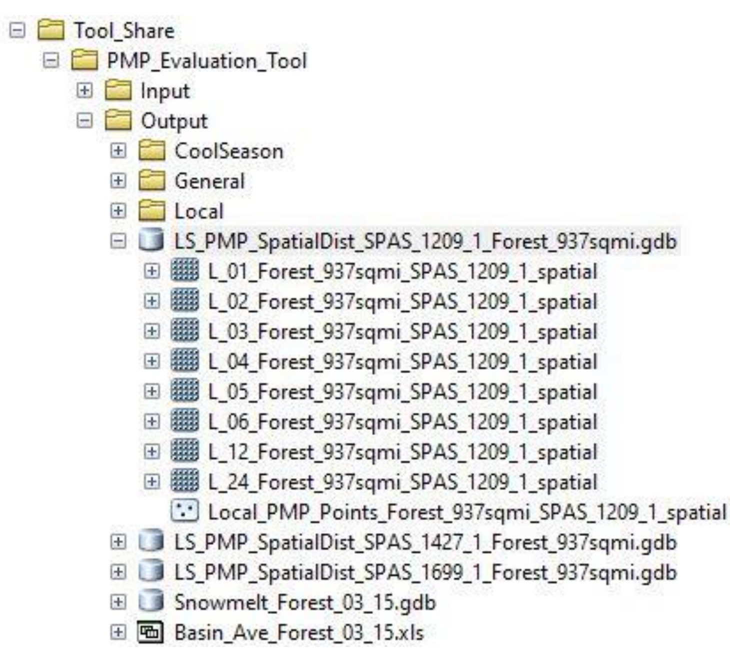
- Finally, if need the Spatial Distribution Tool can be utilized. If needed, double click on the Spatial Distribution Tool script to distribute the PMP depths created earlier based on actual historic storm patterns. In the input dialog below we chose the same Forest Basin file. The location of the PMP_Evaluation_Tool folder is automatically populated. Navigate to the output PMP points created earlier from running the PMP Tool. In this case the example uses local storms. The default spatial location is to center the storm over the basin and to apply the recommended storm patterns. Finally, select an output location.



- Click ok and the tool runs providing feedback like the PMP and Snowmelt Tools. In this case it applies three storm patterns to the default storm pattern created with the PMP tool.

```
Basin Target Location Coordinates:
     X = -97.708
     Y = 48.244
StormPattern = ['LS - Wooster, OH, Jul, 1969 (SPAS_1209_1)', 'LS -
Boyden, IA, Sep, 1926 (SPAS_1427_1)', 'LS - Hayward, WI, Aug, 1941
(SPAS_1699_1)']
StormPattern = LS - Wooster, OH, Jul, 1969 (SPAS_1209_1)

Applying spatial distribution from: LS - Wooster, OH, Jul, 1969 (SPAS_
1209_1) total rainfall pattern
to Local Storm PMP for the Forest drainage basin.

Creating output geodatabase 'LS_PMP_SpatialDist_SPAS_1209_1_Forest_
937sqmi.gdb'
StormNum = SPAS_1209_1

Storm Center Coordinates:
     X = -81.973
     Y = 40.915
Shifting total storm pattern to basin centroid

Determining Spatial Adjustment Factors at grid points...
     Extracting SPAS_1209_1 total storm rainfall depths to
PMP_Points...
     Adding field: SAF_1209_1

Calculating spatially adjusted PMP depths for each field...

     Updating field: PMP_01
          Creating spatially distributed raster:
          Forest_937sqmi_SPAS_1209_1_spatial
```

- Navigate to the output folder location and where the tool has created three new geodatabases with a new set of PMP points and new PMP rasters for each duration based on each storm.



Appendix H
GIS Tool Python Script

Name: Gridded PMP Tool Python Script

Script Version: 1

Python Version: 2.7

ArcGIS Version: ArcGIS Desktop 10.7.1

Author: Applied Weather Associates

Usage: The tool is designed to be executed within an ArcMap environment with an open MXD session.

Description:
    This tool calculates PMP depths for a given drainage basin for the
specified durations. PMP point values are calculated (in inches) for each
grid point (spaced at 90 arc-second intervals) over the project domain. The
points are converted to gridded PMP datasets for each duration.

```
------------------------------------------------------------------------------'''

###############################################################################
## import Python modules

import sys
import arcpy
import os
import traceback
from arcpy import env
import arcpy.analysis as an
import arcpy.management as dm
import arcpy.conversion as con
import numpy as np
import pandas as pd
from pandas import ExcelFile
import matplotlib.pyplot as plt
from heapq import nlargest


env.overwriteOutput = True                          # Set overwrite option
env.addOutputsToMap = False

###############################################################################
## get input parameters

basin = arcpy.GetParameter(0)                              # get AOI Basin Shapefile
home = arcpy.GetParameterAsText(1)                              # get location of 'PMP' Project Folder
outLocation = arcpy.GetParameterAsText(2)
if arcpy.GetParameter(12) == False:
    locDurations = arcpy.GetParameter(3)                         # get local storm durations (string)
    genDurations = arcpy.GetParameter(4)                         # get general storm durations (string)
    coolDurations = arcpy.GetParameter(5)                        # get Cool Season storm durations (string)
else:
    locDurations = ('01','02','03','04','05','06','12','24')
    genDurations = ('01','02','03','04','05','06','12','24','48','72')
    coolDurations = ('01','02','03','04','05','06','12','24','48','72')

weightedAve = arcpy.GetParameter(8)                                        # get option to apply weighted average
(boolean)
#outputTable = arcpy.GetParameter(9)                                       # get file path for basin average
summary table
```

```
includeSubbasin = arcpy.GetParameter(9)                                    # get option add subbasin averages
(boolean)
subbasinIDfield = arcpy.GetParameterAsText(10)                                  # Subbasin ID field from
AOI Basin Shapefile
ddChart = arcpy.GetParameter(11)                                    # get option to create depth-duration
chart(boolean)
runTemporal = arcpy.GetParameter(12)                                        # get option to run temporal
distributions (boolean)

dadGDB = home + "\\Input\\DAD_Tables.gdb"                    # location of DAD tables
adjFactGDB = home + "\\Input\\Storm_Adj_Factors.gdb"              # location of feature datasets containing total
adjustment factors
arcpy.AddMessage("\nDAD Tables geodatabase path:  " + dadGDB)
arcpy.AddMessage("Storm Adjustment Factor geodatabase path:  " + adjFactGDB)

#mxd = arcpy.mapping.MapDocument("CURRENT")
#df = arcpy.mapping.ListDataFrames(mxd)[0]
basAveTables = []                                  # global list of Basin Average Summary tables

def pmpAnalysis(aoiBasin, stormType, durList):

    ###########################################################################
    ## Create PMP Point Feature Class from points within AOI basin and add fields
    def createPMPfc():

        arcpy.AddMessage("\nCreating feature class: 'PMP_Points' in Scratch.gdb...")
        dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid", "vgLayer")          # make a feature layer of
vector grid cells
        dm.SelectLayerByLocation("vgLayer", "INTERSECT", aoiBasin)                       # select the vector grid cells that
intersect the aoiBasin polygon
        dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Grid_Points", "gpLayer")          # make a feature layer of
grid points
        dm.SelectLayerByLocation("gpLayer", "HAVE_THEIR_CENTER_IN", "vgLayer")                  # select the grid points
within the vector grid selection
        con.FeatureClassToFeatureClass("gpLayer", env.scratchGDB, "PMP_Points")                 # save feature layer as
"PMP_Points" feature class
        arcpy.AddMessage("(" + str(dm.GetCount("gpLayer")) + " grid points will be analyzed)\n")

        # Add PMP Fields
        for dur in durList:
            arcpy.AddMessage("\t...adding field: PMP_" + str(dur))
            dm.AddField(env.scratchGDB + "\\PMP_Points", "PMP_" + dur, "DOUBLE")

        # Add STORM Fields (this string values identifies the driving storm by SPAS ID number)
        for dur in durList:
            arcpy.AddMessage("\t...adding field: STORM_" + str(dur))
            dm.AddField(env.scratchGDB + "\\PMP_Points", "STORM_" + dur, "TEXT", "", "", 16, "Storm ID " + dur + "-hour")

        # Add STNAME Fields (this string values identifies the driving storm by SPAS ID number)
#        for dur in durList:
#            arcpy.AddMessage("\t...adding field: STNAME_" + str(dur))
#            dm.AddField(env.scratchGDB + "\\PMP_Points", "STNAME_" + dur, "TEXT", "", "", 50, "Storm Name " + dur + "-
hour")

        return

    ###########################################################################
    ##  Define getAOIarea() function:
    ##  getAOIarea() calculates the area of AOI (basin outline) input shapefile/
    ##  featureclass.  The basin outline shapefile must be projected.  The area
    ##  is sqaure miles, converted from the basin layers projected units (feet
    ##  or meters).  The aoiBasin feature class should only have a single feature
```

```
  ##  (the basin outline).  If there are multiple features, the area will be stored
  ##  for the final feature only.

  def getAOIarea():
    sr = arcpy.Describe(aoiBasin).SpatialReference                    # Determine aoiBasin spatial reference system
    srname = sr.name
    srtype = sr.type
    srunitname = sr.linearUnitName                          # Units
    arcpy.AddMessage("\nAOI basin spatial reference:  " + srname + "\nUnit type: " + srunitname + "\nSpatial reference type: "
+ srtype)

    aoiArea = 0.0
    rows = arcpy.SearchCursor(aoiBasin)
    for row in rows:
      feat = row.getValue("Shape")
      aoiArea += feat.area
    if srtype == 'Geographic':                # Must have a surface projection.  If one doesn't exist it projects a temporary
file and uses that.
      arcpy.AddMessage("\n***The basin shapefile's spatial reference 'Geographic' is not supported.  Projecting temporary
shapefile for AOI.***")
      arcpy.Project_management(aoiBasin,env.scratchGDB + "\\TempBasin",102039)    # Projects AOI Basin (102039 =
USA_Contiguous_Albers_Equal_Area_Conic_USGS_version)
      TempBasin = env.scratchGDB + "\\TempBasin"                         # Path to temporary basin created
in scratch geodatabase
      sr = arcpy.Describe(TempBasin).SpatialReference                  # Determine Spatial Reference of temporary
basin
      aoiArea = 0.0
      rows = arcpy.SearchCursor(TempBasin)                               # Assign area size in square
meters
      for row in rows:
        feat = row.getValue("Shape")
        aoiArea += feat.area
      aoiArea = aoiArea * 0.000000386102                    # Converts square meters to square miles
    elif srtype == 'Projected':
      if srunitname == "Meter":
        aoiArea = aoiArea * 0.000000386102                    # Converts square meters to square miles
      elif srunitname == "Foot" or "Foot_US":
        aoiArea = aoiArea * 0.00000003587                     # Converts square feet to square miles
      else:
        arcpy.AddMessage("\nThe basin shapefile's unit type '" + srunitname + "' is not supported.")
        sys.exit("Invalid linear units")                    # Units must be meters or feet

    aoiArea = round(aoiArea, 3)
    arcpy.AddMessage("\nArea of interest: " + str(aoiArea) + " square miles.")

    if arcpy.GetParameter(6) == False:
      aoiArea = arcpy.GetParameter(7)                       # Enable a constant area size
    aoiArea = round(aoiArea, 1)
    arcpy.AddMessage("\n***Area used for PMP analysis: " + str(aoiArea) + " sqmi***")
    return aoiArea

  ###########################################################################
  ##  Define dadLookup() function:
  ##  The dadLookup() function determines the DAD value for the current storm
  ##  and duration according to the basin area size.  The DAD depth is interpolated
  ##  linearly between the two nearest areal values within the DAD table.
  def dadLookup(stormLayer, duration, area):          # dadLookup() accepts the current storm layer name (string), the
current duration (string), and AOI area size (float)
    #arcpy.AddMessage("\t\tfunction dadLookup() called.")
    durField = "H_" + duration                  # defines the name of the duration field (eg., "H_06" for 6-hour)
    dadTable = dadGDB + "\\" + stormLayer
    rows = arcpy.SearchCursor(dadTable)
```

```
        try:
            row = rows.next()                              # Sets DAD area x1 to the value in the first row of the DAD table.
            x1 = row.AREASQMI
            y1 = row.getValue(durField)
            xFlag = "FALSE"                                # xFlag will remain false for basins that are larger than the largest DAD area.
        except RuntimeError:                               # return if duration does not exist in DAD table
            return

        row = rows.next()
        i = 0
        while row:                                         # iterates through the DAD table - assiging the bounding values directly above and
below the basin area size
            i += 1
            if row.AREASQMI < area:
                x1 = row.AREASQMI
                y1 = row.getValue(durField)
            else:
                xFlag = "TRUE"                             # xFlag is switched to "TRUE" indicating area is within DAD range
                x2 = row.AREASQMI
                y2 = row.getValue(durField)
                break

            row = rows.next()
        del row, rows, i

        if xFlag == "FALSE":
            x2 = area                                      # If x2 is equal to the basin area, this means that the largest DAD area is smaller than
the basin and the resulting DAD value must be extrapolated.
            arcpy.AddMessage("\t\tThe basin area size: " + str(area) + " sqmi is greater than the largest DAD area: " + str(x1) + "
sqmi.\n\t\tDAD value is estimated by extrapolation.")
            y = x1 / x2 * y1                               # y (the DAD depth) is estimated by extrapolating the DAD area to the basin area
size.
            return y                                       # The extrapolated DAD depth (in inches) is returned.

        # arcpy.AddMessage("\nArea = " + str(area) + "\nx1 = " + str(x1) + "\nx2 = " + str(x2) + "\ny1 = " + str(y1) + "\ny2 = " +
str(y2))

        x = area                                           # If the basin area size is within the DAD table area range, the DAD depth is
interpolated
        deltax = x2 - x1                                   # to determine the DAD value (y) at area (x) based on next lower (x1) and next
higher (x2) areas.
        deltay = y2 - y1
        diffx = x - x1

        y = y1 + diffx * deltay / deltax

        if x < x1:
            arcpy.AddMessage("\t\tThe basin area size: " + str(area) + " sqmi is less than the smallest DAD table area: " + str(x1) + "
sqmi.\n\t\tDAD value is estimated by extrapolation.")

        return y                                           # The interpolated DAD depth (in inches) is returned.

    ###########################################################################
    ##  Define updatePMP() function:
    ##  This function updates the 'PMP_XX_' and 'STORM_XX' fields of the PMP_Points
    ##  feature class with the largest value from all analyzed storms stored in the
    ##  pmpValues list.
    def updatePMP(pmpValues, stormID, duration):                                   # Accepts four arguments: pmpValues -
largest adjusted rainfall for current duration (float list); stormID - driver storm ID for each PMP value (text list); and duration
(string)
        pmpfield = "PMP_" + duration
```

```
    stormfield = "STORM_" + duration
    stormTextField = "STNAME_" + duration

    gridRows = env.scratchGDB + "\\PMP_Points"                              # iterates through PMP_Points rows
    i = 0
    with arcpy.da.UpdateCursor(gridRows, (pmpfield, stormfield)) as cursor:
        for row in cursor:
            row[0] = pmpValues[i]                                  # Sets the PMP field value equal to the Max Adj.
Rainfall value (if larger than existing value).
            row[1] = stormID[i]                                    # Sets the storm ID field to indicate the driving storm
event
            cursor.updateRow(row)
            i += 1
    del row, gridRows, pmpfield, stormfield, i
    arcpy.AddMessage("\n\t" + duration + "-hour PMP values update complete. \n")
    return

  ############################################################################
  ##  The outputPMP() function produces raster GRID files for each of the PMP durations.
  ##  Aslo, a space-delimited PMP_Distribition.txt file is created in the 'Text_Output' folder.
  def outputPMP(type, area, outPath):
    desc = arcpy.Describe(basin)
    basinName = desc.baseName
    pmpPoints = env.scratchGDB + "\\PMP_Points"                 # Location of 'PMP_Points' feature class which will
provide data for output

    outType = type[:1]
    outArea = str(int(round(area,0))) + "sqmi"
    outGDB = "PMP_"+ basinName + "_" + outArea +".gdb"
    if not arcpy.Exists(outPath + "\\" + outGDB):                # Check to see if PMP_XXXXX.gdb already exists
        arcpy.AddMessage("\nCreating output geodatabase '" + outGDB + "'")
        dm.CreateFileGDB(outPath, outGDB)
    arcpy.AddMessage("\nCopying PMP_Points feature class to " + outGDB + "...")
    con.FeatureClassToFeatureClass(pmpPoints, outPath + "\\" + outGDB, type + "_PMP_Points_" + basinName + "_" +
outArea)
    pointFC = outPath + "\\" + outGDB + "\\" + type + "_PMP_Points_" + basinName + "_" + outArea
    # addLayerMXD(pointFC)  # calls addLayerMDX function to add output to ArcMap session

    arcpy.AddMessage("\nBeginning PMP Raster Creation...")

    for dur in durList:                                 # This code creates a raster GRID from the current PMP point layer
        durField = "PMP_" + dur
        outLoc = outPath + outGDB +"\\" + outType + "_" + dur + "_" + basinName + "_" + outArea
        arcpy.AddMessage("\n\tInput Path: " + pmpPoints)
        arcpy.AddMessage("\tOutput raster path: " + outLoc)
        arcpy.AddMessage("\tField name: " + durField)
        con.FeatureToRaster(pmpPoints, durField, outLoc, "0.025")
        arcpy.AddMessage("\tOutput raster created...")
    del durField, outLoc, dur

    arcpy.AddMessage("\nPMP Raster Creation complete.")

    if includeSubbasin:                                 # Begin subbasin average calculations
        subbasinID = []
        with arcpy.da.SearchCursor(basin, subbasinIDfield) as cursor:   # Create list of subbasin ID names
            for row in cursor:
                subbasinID.append(row[0])

        subIDtype = arcpy.ListFields(basin, subbasinIDfield)[0].type    # Define the datatype of the subbasin ID field

        if subIDtype != "String":                        # Convert subbasin IDs to a string, if they are not already
            subbasinID = [str(i) for i in subbasinID]
```

```
        subNameLen = max(map(len, subbasinID))                      # Define the length of the longest subbasin ID

        # arcpy.AddMessage("\nList of subbasins...\n" + "\n".join(subbasinID))

        arcpy.AddMessage("\nCreating Subbasin Summary Table...")
        tableName = type + "_PMP_Subbasin_Average" + "_" + outArea
        tablePath = outPath + "\\" + outGDB + "\\" + tableName
        dm.CreateTable(outPath + "\\" + outGDB, tableName)             # Create blank table

        dm.AddField(tablePath, "STORM_TYPE", "TEXT", "", "", 10, "Storm Type")         # Create "Storm Type" field
        dm.AddField(tablePath, "SUBBASIN", "TEXT", "", "", subNameLen, "Subbasin")      # Create "Subbasin" field

        cursor = arcpy.da.InsertCursor(tablePath, "SUBBASIN")          # Create Insert cursor and add a blank row to the table for
each subbasin
        for sub in subbasinID:
            cursor.insertRow([sub])
        del cursor, sub

        dm.CalculateField(tablePath, "STORM_TYPE", '"' + type + '"', "PYTHON_9.3")      # populate storm type field

        i = 0
        for field in arcpy.ListFields(pmpPoints, "PMP_*"):             # Add fields for each PMP duration and calculate the
subbasin averages
            fieldName = field.name
            arcpy.AddMessage("\n\tCalculating subbasin average for " + fieldName + " (weighted)...\n")
            dm.AddField(tablePath, fieldName, "DOUBLE", "", 2)         # Add duration field
            subAveList = []
            for subbasin in subbasinID:                                # Loop through each subbasin
                if subIDtype != "String":                              # Define an SQL expression that specifies the current subbasin
                    sql_exp = """{0} = {1}""".format(arcpy.AddFieldDelimiters(basin, subbasinIDfield), subbasin)
                else:
                    sql_exp = """{0} = '{1}'""".format(arcpy.AddFieldDelimiters(basin, subbasinIDfield), subbasin)
                dm.MakeFeatureLayer(basin, "subbasinLayer", sql_exp)
                outLayer = outPath + "\\" + outGDB + "\\subbasin_" + str(subbasin)
                subBasAve = basinAve("subbasinLayer", fieldName)       # Call the basAve() function passing the subbasin and
duration field
                arcpy.AddMessage("\tSubbasin average for " + str(subbasin) + ":  " + str(subBasAve) + "")
                subAveList.append(subBasAve)                           # Add subbasin average to list
            p = 0
            with arcpy.da.UpdateCursor(tablePath, fieldName) as cursor: # Update the subbasin average summary table with the
subbasin averages
                for row in cursor:
                    row = subAveList[p]
                    cursor.updateRow([row])
                    p += 1

   ##           dm.CalculateField(tablePath, fieldName, fieldAve, "PYTHON_9.3")       # Assigns the basin average
   ##           dur = durList[i]                                       # following lines add alias field names to basin average table
(ArcGIS 10.2.1 or later)
   ##           if dur[0] == "0":
   ##               dur = dur[1:]
   ##           fieldAlias = dur + "-hour PMP"
   ##           dm.AlterField(tablePath, fieldName, "#", fieldAlias)
            i += 1
        arcpy.AddMessage("\nSubbasin summary table complete.")

    arcpy.AddMessage("\nCreating Basin Summary Table...")
    tableName = type + "_PMP_Basin_Average" + "_" + outArea
    tablePath = outPath + "\\" + outGDB + "\\" + tableName
    dm.CreateTable(outPath + "\\" + outGDB, tableName)          # Create blank table
    cursor = arcpy.da.InsertCursor(tablePath, "*")              # Create Insert cursor and add a blank row to the table
```

```
    cursor.insertRow([0])
    del cursor

    dm.AddField(tablePath, "STORM_TYPE", "TEXT", "", "", 30, "Storm Type")        # Create "Storm Type" field
    dm.CalculateField(tablePath, "STORM_TYPE", "'" + type + "'", "PYTHON_9.3")      # populate storm type field

    i = 0
    for field in arcpy.ListFields(pmpPoints, "PMP_*"):        # Add fields for each PMP duration and calculate the basin
average
        fieldName = field.name
        fieldAve = basinAve(basin, fieldName)                # Calls the basinAve() function - returns the average (weighted or not)
        dm.AddField(tablePath, fieldName, "DOUBLE", "", 2)      # Add duration field
        dm.CalculateField(tablePath, fieldName, fieldAve, "PYTHON_9.3")      # Assigns the basin average
##          dur = durList[i]                              # following lines add alias field names to basin average table (ArcGIS 10.2.1 or
later)
##          if dur[0] == "0":
##              dur = dur[1:]
##          fieldAlias = dur + "-hour PMP"
##          dm.AlterField(tablePath, fieldName, "#", fieldAlias)
        i += 1
    arcpy.AddMessage("\nSummary table complete.")
    basAveTables.append(tablePath)

##                      The following lines export a .png image depth duration chart and PMP summary excel file to the output
folder
    if ddChart:
        xValues = durList                                       #Get list of durations for chart
        xValues = [int(i) for i in xValues]                     #Convert duration list to integers
        ax1 = plt.subplot2grid((1,1), (0,0))           #Create variable for subplot in chart
        yValues = []
        pmpFields = [field.name for field in arcpy.ListFields(tablePath, "PMP_*")] # Selects PMP fields for yValues
        with arcpy.da.SearchCursor(tablePath, pmpFields) as cursor:              # Adds PMP depths to yValues
            yValues = next(cursor)
        del cursor, pmpFields

        stormFields = [field.name for field in arcpy.ListFields(pmpPoints, "Storm_*")] # Selects Controlling Storm fields
        contStorms = []                          # List of controlling storms for a single duration
        listOfContStorms = []                    # List of controlling storms for all durations (list of lists)
        i = 0                        # iterator (for "Storm_*" fields)
        while i < len(stormFields):                  # iterates through controlling storm fields
            with arcpy.da.SearchCursor(pmpPoints, stormFields) as cursor:    # Search cursor returns list of unique controlling
storms
                contStorms = sorted({row[i] for row in cursor})
            listOfContStorms.append(contStorms)                       # Add unique storms for current duration to list of controlling
stomrs
            i += 1
        del cursor

        plt.plot(xValues,yValues)                       #Creates chart
        plt.xlabel('Storm Duration in Hours')
        plt.ylabel('Rainfall Depth in Inches')
        plt.title(basinName + " (" + outArea + ") " + type + ' Storm Basin Average PMP\nDepth Duration Chart')
        ax1.grid(True)                                          #Creates grid lines in chart
        yTop = max(yValues) + 1
        ax1.set_ylim(top = yTop)                        #Sets y axis values to match depths +1 1
        ax1.set_xticks(xValues)                         #Sets x axis values to match durations
##          i = 0
##          xy = zip(xValues, yValues)
##          while i < len(stormFields):                  # iterates through controlling storm fields
##              pointXY = xy[i]
##              yLabel = '{0:.1f}'.format(yValues[i])       # round PMP depth to 1 decimal and convert to string
##              stormLabel = str(listOfContStorms[i])      # convert controlling storm ID(s) to string
```

```
##              stormLabel = stormLabel.replace("u", "")    # remove unicode "u"
##              stormLabel = stormLabel.replace("'", "")    # remove unicode ","
##              stormLabel = stormLabel.replace("[", "")    # remove unicode "["
##              stormLabel = stormLabel.replace("]", "")    # remove unicode "]"
##              #ax1.annotate(yLabel + "'\n' + stormLabel, xy=xy[i], textcoords='offset points', size=8, annotation_clip=True)
##              ax1.annotate(yLabel + "'\n' + stormLabel, xy=xy[i], textcoords='data', size=8, annotation_clip=True)
##              i += 1
##          del xy

        plt.savefig(outPath + "\\" + basinName + "_" + type + "_Depth_Duration_Chart.png")          #Save image
        plt.close()                               #Close chart to remove from memory
        arcpy.AddMessage("\nDepth Duration Chart exported to output folder.")
        del xValues, yValues, #df, dfLimited
        return
    return

  ###########################################################################
  ##  The basin() returns the basin average PMP value for a given duration field.
  ##  If the option for a weighted average is checked in the tool parameter the script
  ##  will weight the grid point values based on proportion of area inside the basin.
  def basinAve(aoiBasin, pmpField):
    pmpPoints = env.scratchGDB + "\\PMP_Points"                                         # Path of 'PMP_Points' scratch feature
class
    if weightedAve:
        #arcpy.AddMessage("\tCalculating sub-basin average for " + pmpField + "(weighted)...")
        vectorGridClip = env.scratchGDB + "\\VectorGridClip"                             # Path of 'VectorGridClip' scratch
feature class

        dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid", "vgLayer")             # make a feature
layer of vector grid cells
        dm.SelectLayerByLocation("vgLayer", "INTERSECT", aoiBasin)                            # select the vector grid cells that
intersect the aoiBasin polygon

        an.Clip("vgLayer", aoiBasin, vectorGridClip)                                             # clips aoi vector grid to
basin
        dm.AddField(pmpPoints, "WEIGHT", "DOUBLE")                                        # adds 'WEIGHT' field to
PMP_Points scratch feature class
        dm.MakeFeatureLayer(vectorGridClip, "vgClipLayer")                                   # make a feature layer of basin
clipped vector grid cells
        dm.MakeFeatureLayer(pmpPoints, "pmpPointsLayer")                                   # make a feature layer of
PMP_Points feature class

        dm.AddJoin("pmpPointsLayer", "ID", "vgClipLayer", "ID")                              # joins PMP_Points and
vectorGridBasin tables
        dm.CalculateField("pmpPointsLayer", "WEIGHT", "!vectorGridClip.Shape_Area!", "PYTHON_9.3")               #
Calculates basin area proportion to use as weight for each grid cell.
        dm.RemoveJoin("pmpPointsLayer", "vectorGridClip")

        dm.SelectLayerByLocation("pmpPointsLayer", "INTERSECT", "vgLayer")

        na = arcpy.da.TableToNumPyArray("pmpPointsLayer",(pmpField, 'WEIGHT'))                 # Assign pmpPoints
values and weights to Numpy array (na)
        wgtAve = np.average(na[pmpField], weights=na['WEIGHT'])                            # Calculate weighted average with
Numpy average
        del na
        return round(wgtAve, 2)

    else:
        if includeSubbasin:
            #arcpy.AddMessage("\tCalculating sub-basin average for " + pmpField + "(non-weighted)...")
            vectorGridClip = env.scratchGDB + "\\VectorGridClip"                             # Path of 'VectorGridClip' scratch
feature class
```

```
            dm.MakeFeatureLayer(home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid", "vgLayer")              # make a feature
layer of vector grid cells
            dm.SelectLayerByLocation("vgLayer", "INTERSECT", aoiBasin)                           # select the vector grid cells
that intersect the aoiBasin polygon

            dm.MakeFeatureLayer(pmpPoints, "pmpPointsLayer")                                    # make a feature layer of
PMP_Points feature class

            dm.SelectLayerByLocation("pmpPointsLayer", "INTERSECT", "vgLayer")

            na = arcpy.da.TableToNumPyArray("pmpPointsLayer", pmpField)                         # Assign pmpPoints values
and weights to Numpy array (na)
            fieldAve = np.average(na[pmpField])                                         # Calculates aritmetic mean
            del na
            return round(fieldAve, 2)

         else:
            arcpy.AddMessage("\tCalculating basin average for " + pmpField + "(not weighted)...")
            na = arcpy.da.TableToNumPyArray(pmpPoints, pmpField)                                # Assign pmpPoints values to
Numpy array (na)
            fieldAve = np.average(na[pmpField])                                         # Calculates aritmetic mean
            del na
            return round(fieldAve, 2)

   ##########################################################################
   ##  This basinZone() function returns a list containing transposition zone ID
   ##  (as an integer)


   def      basinZone(bas):       ## This function returns the basin location transposition zone
      tempBasin = env.scratchGDB + "\\tempBasin"
      tempCentroid = env.scratchGDB + "\\tempCentroid"
      joinFeat = home + "\\Input\\Non_Storm_Data.gdb\\Vector_Grid"
      joinOutput = env.scratchGDB + "\\joinOut"
      dm.Dissolve(bas, tempBasin)
      desc = arcpy.Describe(tempBasin)
      sr = desc.spatialReference
      #dm.FeatureToPoint(tempBasin, tempCentroid, "INSIDE")

      dm.CreateFeatureclass(env.scratchGDB,"tempCentroid","POINT",spatial_reference = sr)
      with arcpy.da.InsertCursor(tempCentroid, "SHAPE@XY") as iCur:
         with arcpy.da.SearchCursor(tempBasin,"SHAPE@") as sCur:
            for sRow in sCur:
               cent = sRow[0].centroid         # get the centroid
               iCur.insertRow([(cent.X,cent.Y)])# write it to the new feature class

      an.SpatialJoin(tempCentroid, joinFeat, joinOutput)
      centZone = arcpy.da.SearchCursor(joinOutput, ("TRANS_ZONE",)).next()[0]
      del tempBasin, tempCentroid, joinFeat, joinOutput, desc, sr
      return (centZone)

   ##########################################################################
   ##  The temporalDist() functions applies the temporal distributions scenarios
   ##  to PMP.


   def temporalDistControlStorm_06hr(stormType, outPath, location, areaSize, basinName):                     # Local Storm 6-hr
Temporal Distributions Function

      basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize
```

```
    basinPMPPoints = outPath + "\\" + stormType + "_PMP_Points_" + basinName + "_" + areaSize                    #
Location of basin average PMP table
    controlStormTable = home +
"\\Input\\Non_Storm_Data.gdb\\CONTROLLING_STORM_TEMPORAL_DISTRIBUTIONS_06"
    arcpy.AddMessage(stormType + " Storm - " + dur + "-hour Controlling Storm PMP Temporal Distributions***")
    outTable = outPath + "\\Controlling_Storms_Temporal_Distributions_" + dur
    pointsArray = arcpy.da.TableToNumPyArray(basinPMPPoints, "Storm_" + dur)
    arrayList = []
    for r in pointsArray:
      arrayList.append(r[0])
    distributionList = np.unique(arrayList).tolist()
    controlPatterns = [f.name for f in arcpy.ListFields(controlStormTable)]
    TF = any(item in distributionList for item in controlPatterns)
    if TF == True:
      map = arcpy.FieldMappings()
      fm = arcpy.FieldMap()
      fm.addInputField(controlStormTable, "TIMESTEP")
      map.addFieldMap(fm)
      fm2 = arcpy.FieldMap()
      fm2.addInputField(controlStormTable, "MINUTE")
      map.addFieldMap(fm2)
      for field in distributionList:
        fm3 = arcpy.FieldMap()
        fm3.addInputField(controlStormTable, field)
        map.addFieldMap(fm3)
      arcpy.AddMessage("\n\tCreating temporal distribution table:...")
      arcpy.TableToTable_conversion(controlStormTable, outPath, "Controlling_Storms_Temporal_Distributions_" + dur, "",
map)                    # Copy 6-hour temporal dist. factors table to output location
      sixHour = arcpy.da.SearchCursor(basinPMP, ("PMP_06",)).next()[0]          # Gets 6-hour PMP depth
      for distribution in distributionList:                 # Loops thourgh each 6-hour temporal distribution
        arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
        with arcpy.da.UpdateCursor(outTable, distribution) as cursor:          # Cursor to apply temporal factor to 6-hour
PMP
          for row in cursor:
            row[0] = round(row[0] * sixHour, 3)
            cursor.updateRow(row)
          del row, cursor

      dists6hr = []     # add suffix to distribution pattern name
      for dist in distributionList:
        dists6hr.append(dist + " (6-hr)")

      checkTemporal(stormType, outPath, outTable, dists6hr, dur, areaSize)
    else:
      arcpy.AddMessage("***Controlling Storm does not have any temporal distributions for this duration***")

    return


  def temporalDistControlStorm_24hr(stormType, outPath, location, areaSize, basinName):                # Local Storm 6-hr
Temporal Distributions Function
    basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize
    basinPMPPoints = outPath + "\\" + stormType + "_PMP_Points_" + basinName + "_" + areaSize                    #
Location of basin average PMP table
    controlStormTable = home +
"\\Input\\Non_Storm_Data.gdb\\CONTROLLING_STORM_TEMPORAL_DISTRIBUTIONS_24"

    arcpy.AddMessage(stormType + " Storm - " + dur + "-hour Controlling Storm PMP Temporal Distributions***")
    outTable = outPath + "\\Controlling_Storms_Temporal_Distributions_" + dur
    pointsArray = arcpy.da.TableToNumPyArray(basinPMPPoints, "Storm_" + dur)
    arrayList = []
    for r in pointsArray:
```

```
            arrayList.append(r[0])
        distributionList = np.unique(arrayList).tolist()
        controlPatterns = [f.name for f in arcpy.ListFields(controlStormTable)]
        TF = any(item in distributionList for item in controlPatterns)
        if TF == True:
            map = arcpy.FieldMappings()
            fm = arcpy.FieldMap()
            fm.addInputField(controlStormTable, "TIMESTEP")
            map.addFieldMap(fm)
            fm2 = arcpy.FieldMap()
            fm2.addInputField(controlStormTable, "MINUTE")
            map.addFieldMap(fm2)
            for field in distributionList:
                fm3 = arcpy.FieldMap()
                fm3.addInputField(controlStormTable, field)
                map.addFieldMap(fm3)
            arcpy.AddMessage("\n\tCreating temporal distribution table:...")
            arcpy.TableToTable_conversion(controlStormTable, outPath, "Controlling_Storms_Temporal_Distributions_" + dur, "",
map)                              # Copy 6-hour temporal dist. factors table to output location
            twentyfourHour = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]                  # Gets 6-hour PMP depth
            for distribution in distributionList:                              # Loops thourgh each 6-hour temporal distribution
                arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
                with arcpy.da.UpdateCursor(outTable, distribution) as cursor:                  # Cursor to apply temporal factor to 6-hour
PMP
                    for row in cursor:
                        row[0] = round(row[0] * twentyfourHour, 3)
                        cursor.updateRow(row)
                    del row, cursor

            dists24hr = []             # add suffix to distribution pattern name
            for dist in distributionList:
                dists24hr.append(dist + " (24-hr)")

            checkTemporal(stormType, outPath, outTable, dists24hr, dur, areaSize)
        else:
            arcpy.AddMessage("***Controlling Storm does not have any temporal distributions for this duration***")

        return


    def temporalDistLS2(stormType, outPath, location, areaSize):                      # Local Storm 2-hr Temporal Distributions
Function
        basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize

        arcpy.AddMessage("\n***Local Storm - 2-hour PMP Temporal Distributions***")

        temporalDistTable = home + "\\Input\\Non_Storm_Data.gdb\\LS_TEMPORAL_DISTRIBUTIONS_02hr"                    # LS
2hr Temporal distribution factors tables
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable, "LS_2_*")]                    # Create a list of 2-
hour distribution field names
        outTable = outPath + "\\LS_Temporal_Distributions_2hr"
        arcpy.AddMessage("\n\tCreating 2-hour temporal distribution table:...")
        dm.Copy(temporalDistTable, outTable)                                                                     #
Copy 2-hour temporal dist. factors table to output location
        largestHour = arcpy.da.SearchCursor(basinPMP, ("PMP_01",)).next()[0]                                       #
Calculate largest hour PMP
        secondLargestHour = arcpy.da.SearchCursor(basinPMP, ("PMP_02",)).next()[0] - largestHour              # Calculate 2nd-
largest hour PMP
        arcpy.AddMessage("\n\tLargest Hour: " + str(largestHour))
        arcpy.AddMessage("\tSecond Largest Hour: " + str(secondLargestHour))
        for distribution in distributionList:                                                                    # Loops
thourgh each 2-hour temporal distribution
```

```
        arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
        if distribution == "LS_2_hour_Center":
            accumPMP = 0
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:                    # Cursor to evenly
distribute half of 2nd largest hour over first 6 timesteps
                for row in cursor:
                    if row[1] <= 6:                                                                          # Leave
loop once a row containing a temporal dist. factor (ie, second 24h period) is reached
                        accumPMP +=  secondLargestHour / 12
                        #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:                    # Cursor to apply
temporal factor to 1-hour PMP (steps 7-18)
                for row in cursor:
                    if row[1] > 6 and row[1] <= 18:                                                          #
Constrain update to rows 19-24 (second hour)
                        accumPMP = round((largestHour * row[0]) + (secondLargestHour / 2), 3)
                        #arcpy.AddMessage("\tAccumulated Rain: " + str(accumPMP))
                        row[0] = accumPMP
                        cursor.updateRow(row)
                del row, cursor
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:                     # Cursor to evenly
distribute half of 2nd largest hour over last 6 timesteps
                for row in cursor:
                    accumPMP +=  secondLargestHour / 12
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause
            #arcpy.AddMessage("\nCompleted temporal distribution for: " + distribution)
        else:
            arcpy.AddMessage("\t\tFirst hour...")
            with arcpy.da.UpdateCursor(outTable, distribution) as cursor:                                    # Cursor
to apply temporal factor to 1-hour PMP
                for row in cursor:
                    if not row[0]:
                        #arcpy.AddMessage("\tLeaving loop...")
                        break
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(row[0] * largestHour,3)))
                    row[0] = round(row[0] * largestHour,3)
                    cursor.updateRow(row)
                del row, cursor
            arcpy.AddMessage("\t\tSecond hour...")
            accumPMP = largestHour
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:                     # Cursor
to evenly distribute 2nd largest hour
                for row in cursor:
                    accumPMP += secondLargestHour / 12
                    #arcpy.AddMessage("\tAccumulated Rain: " + str(round(accumPMP, 3)))
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause

    checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)

    del distribution, distributionList, largestHour, secondLargestHour
```

```
        return


    def temporalDistLS6(stormType, outPath, location, areaSize):                    # Local Storm 6-hr Temporal Distributions
Function
        basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize                          # Location of basin
average PMP table

        if stormType == "Local":
            arcpy.AddMessage("\n***Local Storm - 6-hour PMP Temporal Distributions***")

            temporalDistTable_6hr = home + "\\Input\\Non_Storm_Data.gdb\\LS_TEMPORAL_DISTRIBUTIONS_06HR"  # 6-hour
Temporal distribution factors table
            outTable = outPath + "\\LS_Temporal_Distributions_6hr"
            arcpy.AddMessage("\n\tCreating temporal distribution table:...")
            dm.Copy(temporalDistTable_6hr, outTable)                              # Copy 6-hour temporal dist. factors table to output
location
            distributionList = [field.name for field in arcpy.ListFields(temporalDistTable_6hr, "LS*")]     # Create a list of 6-hour
distribution field names
            arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))
            sixHour = arcpy.da.SearchCursor(basinPMP, ("PMP_06",)).next()[0]                # Gets 6-hour PMP depth
            for distribution in distributionList:                              # Loops thourgh each 6-hour temporal distribution
                arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
                with arcpy.da.UpdateCursor(outTable, distribution) as cursor:                # Cursor to apply temporal factor to 6-hour
PMP
                    for row in cursor:
                        row[0] = round(row[0] * sixHour, 3)
                        cursor.updateRow(row)
                    del row, cursor
            checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)
        return


    def temporalDist_24hr(stormType, outPath, location, areaSize):                    # General/Cool Season Storm 24-hr Temporal
Distributions Function
        basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize                          # Location of basin
average PMP table

        if stormType == "General":
            arcpy.AddMessage("\n***" + stormType + " Storm - 24hr PMP Temporal Distributions***")
            temporalDistTable_24hr = home + "\\Input\\Non_Storm_Data.gdb\\GS_TEMPORAL_DISTRIBUTIONS_24HR"    #
General Storm Temporal distribution factors table
            outTable = outPath + "\\GS_Temporal_Distributions_24hr"
            arcpy.AddMessage("\n\tCreating temporal distribution table:...")
            dm.Copy(temporalDistTable_24hr, outTable)                              # Copy temporal dist. factors table to output location
            distributionList = [field.name for field in arcpy.ListFields(temporalDistTable_24hr, "GS*")]       # Create a list of 24-
hour distribution field names
            arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))
            twentyfourHour = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]              # Gets 24-hour PMP depth
            for distribution in distributionList:                              # Loops through each 24-hour temporal distribution
                arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
                with arcpy.da.UpdateCursor(outTable, distribution) as cursor:                # Cursor to apply temporal factor to 24-hour
PMP
                    for row in cursor:
                        row[0] = round(row[0] * twentyfourHour, 3)
                        cursor.updateRow(row)
                    del row, cursor
            checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)

        if stormType == "CoolSeason":
            arcpy.AddMessage("\n***" + stormType + " Storm - 24hr PMP Temporal Distributions***")
```

```
        temporalDistTable_24hr = home + "\\Input\\Non_Storm_Data.gdb\\CS_TEMPORAL_DISTRIBUTIONS_24HR"    #
Cool Season Storm Temporal distribution factors table
        outTable = outPath + "\\CS_Temporal_Distributions_24hr"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable_24hr, outTable)                    # Copy temporal dist. factors table to output location
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable_24hr, "CS*")]      # Create a list of 24-hour
distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))
        twentyfourHour = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]            # Gets 24-hour PMP depth
        for distribution in distributionList:                    # Loops thourgh each 24-hour temporal distribution
          arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
          with arcpy.da.UpdateCursor(outTable, distribution) as cursor:            # Cursor to apply temporal factor to 24-hour
PMP
            for row in cursor:
              row[0] = round(row[0] * twentyfourHour, 3)
              cursor.updateRow(row)
            del row, cursor
        checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)
      return


  def temporalDist_48hr(stormType, outPath, location, areaSize):              # General/Cool Season Storm 48-hr Temporal
Distributions Function
      basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize                # Location of basin
average PMP table

      if stormType == "General":
        arcpy.AddMessage("\n***" + stormType + " Storm - 48hr PMP Temporal Distributions***")

        temporalDistTable_48hr = home + "\\Input\\Non_Storm_Data.gdb\\GS_TEMPORAL_DISTRIBUTIONS_48HR"    #
General Storm Temporal distribution factors table
        outTable = outPath + "\\GS_Temporal_Distributions_48hr"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable_48hr, outTable)                    # Copy temporal dist. factors table to output location
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable_48hr, "GS*")]      # Create a list of 48-
hour distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))

        largest24 = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]              # Calculate largest 24-hour period
PMP
        second24 = (arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0] - largest24)/2         # Calculate the next largest
24-hr period PMP and divide by 2

        arcpy.AddMessage("\n\tLargest 24-hour Period: " + str(largest24))
        arcpy.AddMessage("\tFirst 12-hour: " + str(second24))
        arcpy.AddMessage("\tLast 12-hour: " + str(second24))


        for distribution in distributionList:                    # Loops thourgh each 24-hour temporal distribution
          arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
          arcpy.AddMessage("\t\tFirst 12-hour Period...")
          accumPMP = 0
          with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:          # Cursor to evenly distribute
half of 2nd largest 24-hr into first 12 hours
            for row in cursor:
              if row[1] <= 48:                        # Leave loop once a row containing a temporal dist. factor (ie, first 12h
period) is reached
                accumPMP +=  second24 / 48
                row[0] = round(accumPMP, 3)
                cursor.updateRow(row)
            del row, cursor
```

```
            arcpy.AddMessage("\t\tLargest 24-hour Period...")
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:            # Cursor to apply temporal
factors to largest 24-hour PMP
                for row in cursor:
                    if row[1] > 48 and row[1] <= 144:                        # Constrain update to rows 49-144 (second 24hr period)
                        accumPMP = (largest24 * row[0]) + second24
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tLast 12-hour Period...")
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:            # Cursor to evenly distribute half
of 2nd largest 24-hr into last 12 hours
                for row in cursor:
                    accumPMP +=  second24 / 48
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause
        checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)

    if stormType == "CoolSeason":
        arcpy.AddMessage("\n***" + stormType + " Storm - 48hr PMP Temporal Distributions***")

        temporalDistTable_48hr = home + "\\Input\\Non_Storm_Data.gdb\\CS_TEMPORAL_DISTRIBUTIONS_48HR"    #
Cool Season Storm Temporal distribution factors table
        outTable = outPath + "\\CS_Temporal_Distributions_48hr"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable_48hr, outTable)                        # Copy temporal dist. factors table to output location
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable_48hr, "CS*")]       # Create a list of 48-hour
distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))

        largest24 = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]                        # Calculate largest 24-hour period
PMP
        second24 = (arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0] - largest24)/2            # Calculate the third largest
12-hr period PMP and divide by 2


        arcpy.AddMessage("\n\tLargest 24-hour Period: " + str(largest24))
        arcpy.AddMessage("\tFirst 12-hour: " + str(second24))
        arcpy.AddMessage("\tLast 12-hour: " + str(second24))


        for distribution in distributionList:                            # Loops thourgh each 24-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            arcpy.AddMessage("\t\tFirst 12-hour Period...")
            accumPMP = 0
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:            # Cursor to evenly distribute
half of 2nd largest 24-hour PMP to first 12 hours
                for row in cursor:
                    if row[1] <= 48:                        # Leave loop once a row containing a temporal dist. factor (ie, first 12h
period) is reached
                        accumPMP +=  second24 / 48
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tLargest 24-hour Period...")
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:            # Cursor to apply temporal
factors to largest 24-hour PMP
                for row in cursor:
```

```
            if row[1] > 48 and row[1] <= 144:                    # Constrain update to rows 49-144 (second 24hr period)
                accumPMP = (largest24 * row[0]) + second24
                row[0] = round(accumPMP, 3)
                cursor.updateRow(row)
        del row, cursor

        arcpy.AddMessage("\t\tLast 12-hour Period...")
        whereClause = distribution + " IS NULL"
        with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:            # Cursor to evenly distribute half
of 2nd largest 24-hr into last 12 hours
            for row in cursor:
                accumPMP +=  second24 / 48
                row[0] = round(accumPMP, 3)
                cursor.updateRow(row)
            del row, cursor, accumPMP, whereClause

        checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)
    return


def temporalDist_72hr(stormType, outPath, location, areaSize):                    # General/Cool Seaon Storm 72-hr Temporal
Distributions Function
    basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize                        # Location of basin
average PMP table

    if stormType == "General":
        arcpy.AddMessage("\n***" + stormType + " Storm - 72hr PMP Temporal Distributions***")
        temporalDistTable_72hr = home + "\\Input\\Non_Storm_Data.gdb\\GS_TEMPORAL_DISTRIBUTIONS_72HR"    #
General Storm Temporal distribution factors table
        outTable = outPath + "\\GS_Temporal_Distributions_72hr"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable_72hr, outTable)                        # Copy temporal dist. factors table to output location
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable_72hr, "GS*")]       # Create a list of 72-
hour distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))

        largest24 = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]                                    # Calculate
largest 24-hour period PMP
        second24 = arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0] - largest24          # Calculate 2nd-largest 24-hour
period PMP
        third24 = arcpy.da.SearchCursor(basinPMP, ("PMP_72",)).next()[0] - arcpy.da.SearchCursor(basinPMP,
("PMP_48",)).next()[0]  # Calculate 3rd-largest 24-hour period PMP

        arcpy.AddMessage("\n\tLargest 24-hour: " + str(largest24))
        arcpy.AddMessage("\tSecond largest 24-hour: " + str(second24))
        arcpy.AddMessage("\tThird largest 24-hour: " + str(third24))

        for distribution in distributionList:                          # Loops thourgh each 72-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            arcpy.AddMessage("\t\tFirst 24-hour Period...")
            accumPMP = 0
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:           # Cursor to evenly distribute 2nd
largest 24-hour
                for row in cursor:
                    if row[1] <= 96:                          # Leave loop once a row containing a temporal dist. factor (ie, second
24h period) is reached
                        accumPMP +=  second24 / 96
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tSecond 24-hour Period...")
```

```
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:            # Cursor to apply temporal
factors to largest 24-hour PMP
                for row in cursor:
                    if row[1] > 96 and row[1] <= 192:                          # Constrain update to rows 97-192 (second 24hr period)
                        accumPMP = (largest24 * row[0]) + second24
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tThird 24-hour Period...")
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:            # Cursor to evenly distribute 3nd
largest hour over remaining empty rows
                for row in cursor:
                    accumPMP +=  third24 / 96
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause
        checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)

    if stormType == "CoolSeason":
        arcpy.AddMessage("\n***" + stormType + " Storm - 72hr PMP Temporal Distributions***")
        temporalDistTable_72hr = home + "\\Input\\Non_Storm_Data.gdb\\CS_TEMPORAL_DISTRIBUTIONS_72HR"    #
Cool Season Storm Temporal distribution factors table
        outTable = outPath + "\\CS_Temporal_Distributions_72hr"
        arcpy.AddMessage("\n\tCreating temporal distribution table:...")
        dm.Copy(temporalDistTable_72hr, outTable)                         # Copy temporal dist. factors table to output location
        distributionList = [field.name for field in arcpy.ListFields(temporalDistTable_72hr, "CS*")]       # Create a list of 72-hour
distribution field names
        arcpy.AddMessage("\n\tDistribution Field Names: " + str(distributionList))

        largest24 = arcpy.da.SearchCursor(basinPMP, ("PMP_24",)).next()[0]                                        # Calculate
largest 24-hour period PMP
        second24 = arcpy.da.SearchCursor(basinPMP, ("PMP_48",)).next()[0] - largest24          # Calculate 2nd-largest 24-hour
period PMP
        third24 = arcpy.da.SearchCursor(basinPMP, ("PMP_72",)).next()[0] - arcpy.da.SearchCursor(basinPMP,
("PMP_48",)).next()[0]   # Calculate 3rd-largest 24-hour period PMP

        arcpy.AddMessage("\n\tLargest 24-hour: " + str(largest24))
        arcpy.AddMessage("\tSecond largest 24-hour: " + str(second24))
        arcpy.AddMessage("\tThird largest 24-hour: " + str(third24))

        for distribution in distributionList:                            # Loops thourgh each 24-hour temporal distribution
            arcpy.AddMessage("\n\tApplying temporal distribution for: " + distribution)
            arcpy.AddMessage("\t\tFirst 24-hour Period...")
            accumPMP = 0
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:            # Cursor to evenly distribute 2nd
largest hour
                for row in cursor:
                    if row[1] <= 96:                             # Leave loop once a row containing a temporal dist. factor (ie, second
24h period) is reached
                        accumPMP +=  second24 / 96
                        row[0] = round(accumPMP, 3)
                        cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tSecond 24-hour Period...")
            with arcpy.da.UpdateCursor(outTable, [distribution, "TIMESTEP"]) as cursor:            # Cursor to apply temporal
factors to largest 24-hour PMP
                for row in cursor:
                    if row[1] > 96 and row[1] <= 192:                          # Constrain update to rows 97-192 (second 24hr period)
                        accumPMP = (largest24 * row[0]) + second24
```

```
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor

            arcpy.AddMessage("\t\tThird 24-hour Period...")
            whereClause = distribution + " IS NULL"
            with arcpy.da.UpdateCursor(outTable, distribution, whereClause) as cursor:           # Cursor to evenly distribute 3nd
largest hour over remaining empty rows
                for row in cursor:
                    accumPMP +=  third24 / 96
                    row[0] = round(accumPMP, 3)
                    cursor.updateRow(row)
                del row, cursor, accumPMP, whereClause

        checkTemporal(stormType, outPath, outTable, distributionList, dur, areaSize)
    return


##  This portion of the code checks to make sure none of the temporal distributions
##  are exceeding the PMP values for any durations.  It adds a table to the output
##  folder called CheckTemporal.
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


def checkTemporal(stormType, outPath, TemporalTable, distributionFields, dur, areaSize):
    basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + areaSize                    # Location of basin
average PMP table
    pmpFields = [field.name for field in arcpy.ListFields(basinPMP, "PMP_*")]                  # PMP duration run
    temporalFields = [field.name for field in arcpy.ListFields(TemporalTable)]
    table = arcpy.Describe(TemporalTable)
    tableName = table.name


    pmp = []                                                  #Creates empty list and updates with PMP values for
each duration run
    i = 0
    while i < len(pmpFields):
        with arcpy.da.SearchCursor(basinPMP,pmpFields) as cursor:
            for row in cursor:
                pmp.append(row[i])
                i += 1
    del i, cursor

    checkTable = outPath + "\\Temporal_Distribution_Check_" + stormType
    arcpy.AddMessage("\nCheckTable: "  + checkTable)
    maxFields = []                                            #Create Max fields for each duration
    checkFields = []                                          #Create Check fields for each duration
    if arcpy.Exists(checkTable):
        with arcpy.da.InsertCursor(checkTable, "PATTERN") as cursor:
            for val in distributionFields:
                cursor.insertRow([val])
        i = 0                                                 #Populate fields
        for pmpField in pmpFields:
            with arcpy.da.UpdateCursor(checkTable, pmpField) as cursor:
                for row in cursor:
                    row = pmp[i]
                    cursor.updateRow([row])
                i += 1
        del i, cursor
    else:
```

```
        checkTable = dm.CreateTable(outPath, "Temporal_Distribution_Check_" + stormType)                    #Creates table in
output folder, adds field, and populates field with distributions
        dm.AddField(checkTable, "PATTERN", "TEXT", "", "", 50)
        with arcpy.da.InsertCursor(checkTable, "PATTERN") as cursor:
            for val in distributionFields:
                cursor.insertRow([val])
        for maxField in pmpFields:
            newField = maxField.replace("PMP","MAX")
            maxFields.append(newField)
        del newField
        for checkField in pmpFields:
            newField = checkField.replace("PMP","CHECK")
            checkFields.append(newField)
        del newField
        i = 0                                                            #Populate fields
        for pmpField in pmpFields:
            dm.AddField(checkTable, pmpField, "DOUBLE", "", "", 50)
            dm.AddField(checkTable, maxFields[i], "DOUBLE", "", "", 50)
            dm.AddField(checkTable, checkFields[i], "TEXT", "", "", 50)
            with arcpy.da.UpdateCursor(checkTable, pmpField) as cursor:
                for row in cursor:
                    row = pmp[i]
                    cursor.updateRow([row])
                i += 1
        del i, cursor

    step = arcpy.da.SearchCursor(TemporalTable,("MINUTE",)).next()[0]
    if step == 15:
        dic = {"01": 4, "02": 8, "03": 12, "04": 16, "05": 20, "06": 24, "12": 48, "24": 96, "48": 192, "72": 288, "96": 384, "120":
480}  # Dictionary to convert durations into 15-minute timesteps
    elif step == 5:
        dic = {"01": 12, "02": 24, "03": 36, "04": 48, "05": 60, "06": 72, "12": 144, "24": 288, "48": 576, "72": 864, "96": 1152,
"120": 1440}
    elif step == 60:
        dic = {"01": 1, "02": 2, "03": 3, "04": 4, "05": 5, "06": 6, "12": 12, "24": 24, "48": 48, "72": 72, "96": 96, "120": 120}
    # arcpy.AddMessage(str(step) + " Minute distribution Pattern.....")

    maxFields = [field.name for field in arcpy.ListFields(checkTable, "MAX*")]
    i = 0                              # Calculates incremental PMP depths from temporal distribution and gets maximum rainfall
for each duration run
    d = durList.index(dur) + 1
    for dur in durList[:d]:
        k = dic[dur]
        p = 3                                                                                    # Skip first 3 fields in
temporaltable (objectID, Timesteps, minutes)
        for distribution in distributionFields:
            incPMP = []
            previousRow = 0
            with arcpy.da.SearchCursor(TemporalTable, temporalFields) as cursor:
                for row in cursor:
                    increment = row[p] - previousRow
                    previousRow = row[p]
                    incPMP.append(increment)
            na = np.array(incPMP)
            sumList = np.convolve(na,np.ones(k))
            maxPMP = max(sumList)
            maximumPMP = math.trunc(maxPMP * 10 ** 2.0) / 10 ** 2.0
            p += 1

            x = 0
            with arcpy.da.UpdateCursor(checkTable, ["PATTERN", maxFields[i]]) as cursor:             # Updates table with max
values
```

```
                for row in cursor:
                    if row[0] == distribution:
                        row[1] = maximumPMP
                        x += 1
                        cursor.updateRow(row)
            i += 1
        del i, k, cursor, x
        with arcpy.da.UpdateCursor(checkTable, '*') as cursor:                            # Compares PMP values to max values for each
duration.  If PMP values are larger update check field with PASS if not FAIL
            for row in cursor:
                rec = dict(zip(cursor.fields, row))
                arcpy.AddMessage("\n\n\tChecking temporally distributed depth-durations against PMP: " + rec['PATTERN'] + "\n")
                for k, v in rec.items():
                    if not k.startswith('PMP_'):
                        continue
                    _, n = k.split('_')
                    try:
                                    # This try/except skips comparisons for additional durations not present in current temporal pattern
                        mx = rec['MAX_{}'.format(n)]
                        rec['CHECK_{}'.format(n)] = 'FAIL' if v < mx else 'PASS'
                    except:
                        arcpy.AddMessage("\n\tDuration not present...")
                        continue
                    if rec['CHECK_{}'.format(n)] == 'PASS':
                        arcpy.AddMessage("\t" + str(n) + "-hour \n\t\tPMP value is... " + str(v) + "  \n\t\tmax rainfall value is..." + str(mx)
+ "\n\t\tThis distribution.... " + rec['CHECK_{}'.format(n)])
                    else:
                        arcpy.AddMessage("\t" + str(n) + "-hour \n\t\tPMP value is... " + str(v) + "  \n\t\tmax rainfall value is..." + str(mx)
+ "\n\t\tThis distribution.... " + rec['CHECK_{}'.format(n)]+ "\n\t\t**Max values for duration are exceeding PMP values. Use of
this temporal distribtion not recommended.")
                cursor.updateRow([rec[k] for k in cursor.fields])
        del cursor, k, v, rec
        return


    ##########################################################################
    ##  The temporalCritStacked() function applies the critically stacked
    ##  temporal distributions scenarios.  The function accepts the storm type,
    ##  output .gdb path, AOI area size, PMP duration string (hours), and
    ##  integer timestep duration (minutes). The function outputs a gdb table.


    def temporalCritStacked(stormType, outPath, area, duration, timestep):                                                        #
Function applied Critically Stacked temporal distribution
        basinPMP = outPath + "\\" + stormType + "_PMP_Basin_Average_" + area                     # Location of basin average
PMP table
        if stormType == "Local" and duration == "06":                                                                  # These
conditional statements define the field name based on storm type, PMP duration, and timestep duration
            csField = "LS_" + duration + "_HOUR_" + str(timestep) + "MIN_CRIT_STACKED"
        elif stormType == "General":
            csField = "GS_" + duration + "_HOUR_" + str(timestep) + "MIN_CRIT_STACKED"
        elif stormType == "CoolSeason":
            csField = "CS_" + duration + "_HOUR_" + str(timestep) + "MIN_CRIT_STACKED"
        else:
            arcpy.AddMessage("\n***Invalid storm type: " + stormType)
            return
        arcpy.AddMessage("\n***" + duration + "-hour " + str(timestep) + "-min Critically Stacked Temporal Distribution***")
        tableName = "Temporal_Distribution_" + duration + "hr_" + str(timestep) + "min_Crit_Stacked"                  #
Output table name
        tablePath = outPath + "\\" + tableName                                                                        # Output table full path
        pmpFields = [field.name for field in arcpy.ListFields(basinPMP, "PMP*")]                                  # Gets the
"PMP_XX" field names from the basin avereage PMP table
```

```
    if duration == "06":                                                        # These conditional statements define
the key durations needed to build the critically stacked patterns for the following durations...
        keyDurations = [1, 2, 3, 4, 5, 6]
    elif duration == "12":
        keyDurations = [1, 2, 3, 4, 5, 6, 12]
    elif duration == "24":
        keyDurations = [1, 2, 3, 4, 5, 6, 12, 24]
    elif duration == "48":
        keyDurations = [1, 2, 3, 4, 5, 6, 12, 24, 48]
    elif duration == "72":
        keyDurations = [1, 2, 3, 4, 5, 6, 12, 24, 48, 72]
    elif duration == "96":
        keyDurations = [1, 2, 3, 4, 5, 6, 12, 24, 48, 72, 96]
    elif duration == "120":
        keyDurations = [1, 2, 3, 4, 5, 6, 12, 24, 48, 72, 96, 120]
    else:
        arcpy.AddMessage("\n\t...Critically stacked temporal distribution not available for " + duration + "-hour duration.")
        return
    timestepLen = int(duration) * 60 // timestep                                # number of rows in output
table
    xValues = [0]
    for i in keyDurations:                                                      # defines the known x-values (xp) to be
used in the interpolation
        xVal = i * timestepLen / int(duration)
        xValues.append(xVal)
    del i, xVal
    yValues = [0]
    d = 0
    for i in keyDurations:                                                      # defines the known y-values (fp) to be
used in the interpolation
        pmpDepth = arcpy.da.SearchCursor(basinPMP, pmpFields).next()[d]
        yValues.append(pmpDepth)
        d += 1
    del d, i, pmpDepth

    x = np.arange(0, timestepLen + 1, 1)                                        # defines the x points at which to
interpolate values
    xp = np.asarray(xValues)                                                    # np.asarray converts lists into
numpy arrays
    fp = np.asarray(yValues)
    y = np.interp(x, xp, fp)
    inc = []
    prevDepth = 0
    i = 0
    for depth in np.nditer(y):                                                  # populates incremental depths list
'inc' with y array
        inc.append(depth - prevDepth)
        prevDepth = depth
        i += 1
    del i, prevDepth
    periods = int(duration)                                                     # defines number of periods (known
hours) as the duration
    periodLen = 60 // timestep                                                  # defines number of timesteps
(minutes) in each period
    ranks = []
    stackRank = 1
    i = 0
    while i < periods:                                                          # populates list 'ranks' with a rank
integer, one entry per period
        ranks.append(stackRank)
        stackRank += periodLen
```

```
        i += 1
    del i

    orderRanks = []
    orderRanks.insert(0, ranks.pop(0))
    for i in range (timestepLen // periodLen):                    ## orders the ranks according
to critically stacked pattern.  Pulls
        if ranks:                                                  ## (p op()) the first rank from the ranks list
and places it in the orderRanks
            orderRanks.insert(0, ranks.pop(0))                     ## list.  Places next two ranks at
the beginning of the list
        if ranks:                                                  ## and the following at the end of the list.
Repeats until ranks is empty.
            orderRanks.insert(0, ranks.pop(0))
        if ranks:
            orderRanks.append(ranks.pop(0))
    del i
    orderRanks += [orderRanks.pop(0)]
    if orderRanks[0] == max(orderRanks):                           ## Moves last rank to the end
of of orderRanks list.
        arcpy.AddMessage("\n*** moving first rank to last")
        orderRanks.append(orderRanks.pop(max))
    orderInc = []
    n = 0
    for i in range(periods):                                       # gets the nth largest increment where
n is the ordered Rank.
        for q in range(periodLen):
            nthLargest = nlargest(orderRanks[n], inc)[-1]
            orderInc.append(nthLargest)
        n += 1
    del n, i, q
    cumulative = []
    prevInc = 0
    for i in orderInc:                                             # Conve rts the incremental depths to
cumulatove depths and places in cumulative list
        value = round(i + prevInc, 3)
        cumulative.append(value)
        prevInc = i + prevInc
        i += 1
    del i, prevInc
    timesteps = x.tolist()                                         # Converts the timesteps array (x) to a
list then removes the first zero entry
    timesteps.pop(0)
    minutes = []
    minutesInc = timestep
    for i in range(timestepLen):                                   # Constructs the minutes list to be
used in output column based on timestep interval
        minutes.append(minutesInc)
        minutesInc += timestep
    del i
    dm.CreateTable(outPath, tableName)                             # Create the output
geodatabase table
    dm.AddField(tablePath, "TIMESTEP", "DOUBLE")                   # Create
"TIMESTEP" field
    dm.AddField(tablePath, "MINUTES", "DOUBLE")                    # Create "MINUTES"
field
    dm.AddField(tablePath, csField, "DOUBLE")                      # Create cumulated
rainfall field
    zipped = zip(timesteps, minutes, cumulative)                   # Zip up lists of output items.
    fields = ('TIMESTEP', 'MINUTES', csField)                      # Ou tput table field names
    arcpy.AddMessage("\n\tApplying temporal distribution for: " + csField)
```

```
        with arcpy.da.InsertCursor(tablePath, fields) as cursor:                                                        # Cursor to
populate output Critically Stacked table
            for i in zipped:
                cursor.insertRow(i)
        del cursor, i
        return


    ###########################################################################
    ##  This portion of the code iterates through each storm feature class in the
    ##  'Storm_Adj_Factors' geodatabase (evaluating the feature class only within
    ##  the Local, Tropical, or general feature dataset).  For each duration,
    ##  at each grid point within the aoi basin, the transpositionality is
    ##  confirmed.  Then the DAD precip depth is retrieved and applied to the
    ##  total adjustement factor to yield the total adjusted rainfall.  This
    ##  value is then sent to the updatePMP() function to update the 'PMP_Points'
    ##  feature class.
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##


    desc = arcpy.Describe(basin)                                    # Check to ensure AOI input shape is a Polygon. If not - exit.
    basinShape = desc.shapeType
    if desc.shapeType == "Polygon":
        arcpy.AddMessage("\nBasin shape type: " + desc.shapeType)
    else:
        arcpy.AddMessage("\nBasin shape type: " + desc.shapeType)
        arcpy.AddMessage("\nError: Input shapefile must be a polygon!\n")
        sys.exit()

    createPMPfc()                                                   # Call the createPMPfc() function to create the PMP_Points feature
class.

    env.workspace = adjFactGDB                                      # the workspace environment is set to the
'Storm_Adj_Factors' file geodatabase

    aoiSQMI = round(getAOIarea(),2)                                 # Calls the getAOIarea() function to assign area of AOI
shapefile to 'aoiSQMI'
    if aoiSQMI > 100 and stormType is "Local":
        arcpy.AddMessage("\n***Warning - Local storm PMP depths only valid for basins 100 square miles or smaller***")

    stormList = arcpy.ListFeatureClasses("", "Point", stormType)            # List all the total adjustment factor feature classes
within the storm type feature dataset.
    for dur in durList:
        arcpy.AddMessage("\n*****************************************************************\nEvaluating " + dur + "-
hour duration...")

        pmpList = []
        driverList = []
        gridRows = arcpy.SearchCursor(env.scratchGDB + "\\PMP_Points")
        try:
            for row in gridRows:
                pmpList.append(0.0)                                 # creates pmpList of empty float values for each grid point to
store final PMP values
                driverList.append("STORM")                          # creates driverList of empty text values for each grid
point to store final Driver Storm IDs
            del row, gridRows
        except UnboundLocalError:
            arcpy.AddMessage("\n***Error: No data present within basin/AOI area.***\n")
            sys.exit()

        env.workspace = adjFactGDB
```

```
    for storm in stormList[:]:
        arcpy.AddMessage("\n\tEvaluating storm: " + storm + "...")
        arcpy.MakeFeatureLayer_management(storm, "stormLayer")                      # creates a feature layer for the
current storm
        dm.SelectLayerByLocation("stormLayer", "HAVE_THEIR_CENTER_IN", "vgLayer")   # examines only the grid points
that lie within the AOI
        gridRows = arcpy.SearchCursor("stormLayer")
        pmpField = "PMP_" + dur
        i = 0
        try:
            dadPrecip = round(dadLookup(storm, dur, aoiSQMI),3)
            arcpy.AddMessage("\t\t" + dur + "-hour DAD value:  " + str(dadPrecip) + chr(34))
        except TypeError:                                    # In no duration exists in the DAD table - move to the next storm
            arcpy.AddMessage("\t***Duration '" + str(dur) + "-hour' is not present for " + str(storm) + ".***\n")
            continue
        arcpy.AddMessage("\t\tComparing " + storm + " adjusted rainfall values against current driver values...")
        transCounter = 0                                     # Counter for number of grid points transposed to
        for row in gridRows:
            if row.TRANS == 1:                               # Only continue if grid point is transpositionable ('1' is
transpostionable, '0' is not).
                try:                                         # get total adj. factor if duration exists
                    transCounter += 1
                    adjRain = round(dadPrecip * row.TAF,1)
                    if adjRain > pmpList[i]:
                        pmpList[i] = adjRain
                        driverList[i] = storm
                except RuntimeError:
                    arcpy.AddMessage("\t\t   *Warning*  Total Adjusted Raifnall value falied to set for row " + str(row.CNT))
                    break
                del adjRain
            i += 1
        if transCounter == 0:
            arcpy.AddMessage("\t\tStorm not transposable to basin. Removing " + storm + " from list...\n")
            stormList.remove(storm)
        else:
            arcpy.AddMessage("\t\tTransposed to " + str(transCounter) + "/" + str(i) + " grid points...\n")
        del row, transCounter
    del storm, gridRows, dadPrecip
    updatePMP(pmpList, driverList, dur)             # calls function to update "PMP Points" feature class
  del pmpList, stormList

  arcpy.AddMessage("\n'PMP_Points' Feature Class 'PMP_XX' fields update complete for all '" + stormType + "' storms.")

  outputPMP(stormType, aoiSQMI, outputPath)             # calls outputPMP() function
  outArea = str(int(round(aoiSQMI,0))) + "sqmi"
  outGDB = outLocation + "\\" + stormType + "\\PMP_" + desc.baseName + "_" + outArea + ".gdb"
  basinName = desc.baseName

  if runTemporal:                                  #Calls temporal distribution functions
    centroidLocation = basinZone(basin)
    arcpy.AddMessage("\nBasin Centroid Transposition Zone: " + str(centroidLocation))

    for dur in durList:
        if stormType == "Local" and dur == "02":
            temporalDistLS2(stormType, outGDB, centroidLocation, outArea)
        if dur == "06":
            temporalDistLS6(stormType, outGDB, centroidLocation, outArea)
            temporalDistControlStorm_06hr(stormType, outGDB, centroidLocation, outArea, basinName)
        if stormType == "Local" and dur == "06":
            temporalCritStacked(stormType, outGDB, outArea, dur, 5)
        if dur == "24":
            temporalDist_24hr(stormType, outGDB, centroidLocation, outArea)
```

```
            temporalCritStacked(stormType, outGDB, outArea, dur, 15)
            temporalDistControlStorm_24hr(stormType, outGDB, centroidLocation, outArea, basinName)
        if dur == "48":
            temporalDist_48hr(stormType, outGDB, centroidLocation, outArea)
        if dur == "72":
            temporalDist_72hr(stormType, outGDB, centroidLocation, outArea)


    i = 0                                                   #Creates CSV files of all output tables
    csvPath = outLocation + "\\" + stormType + "\\CSV_" + desc.baseName + "_" + outArea + "\\"
    if not arcpy.Exists(outLocation + "\\" + stormType + "\\CSV_" + desc.baseName + "_" + outArea):
        arcpy.CreateFolder_management(outLocation + "\\" + stormType + "\\", "CSV_" + desc.baseName + "_" + outArea)
    arcpy.AddMessage("\n\t...Creating output tables as CSV files.. ")
    env.workspace = outGDB
    outTables = arcpy.ListTables()
    for t in outTables:
        arcpy.TableToTable_conversion(t, csvPath, outTables[i] + ".csv")
        i += 1
    xmlFiles = os.listdir(csvPath)
    for file in xmlFiles:
        if file.endswith(".xml"):
            os.remove(os.path.join(csvPath,file))
    return
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##


def outputBasAveTable():
    arcpy.AddMessage("\nCreating basin average summary table.\n")
    tableList = basAveTables
    for table in tableList:
        arcpy.AddMessage("\t\tMerging tables... " + table)

    dm.Merge(basAveTables, outputTable)
    ##  addLayerMXD(outputTable)  adds output table to ArcMap session

    return
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##


def addLayerMXD(addFC):
    desc = arcpy.Describe(addFC)
    layerName = desc.name
    arcpy.AddMessage("\nAdding " + layerName + " table to current MXD...")
    if desc.dataType == "FeatureClass":
        dm.MakeFeatureLayer(addFC, layerName)
        layer = arcpy.mapping.Layer(layerName)
        arcpy.mapping.AddLayer(df, layer)
        arcpy.AddMessage("\n" + layerName + " added to current map session.\n")
    elif desc.dataType == "Table":
        layer = arcpy.mapping.TableView(desc.catalogPath)
        arcpy.mapping.AddTableView(df, layer)
        arcpy.AddMessage("\n" + layerName + " added to current map session.\n")
    elif desc.dataType == "ArcInfoTable":
        layer = arcpy.mapping.TableView(desc.catalogPath + ".dbf")
        arcpy.mapping.AddTableView(df, layer)
        arcpy.AddMessage("\n" + layerName + " added to current map session.\n")

    del desc, layerName, layer
    return
```

```
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##


if locDurations:
    type = "Local"
    durations = locDurations
    dm.CreateFolder(outLocation, type)
    outputPath = outLocation + "\\Local\\"
    arcpy.AddMessage("\nRunning PMP analysis for storm type: " + type)
    pmpAnalysis(basin, type, durations)         # Calls the pmpAnalysis() function to calculate the local storm PMP
    arcpy.AddMessage("\nLocal storm analysis
complete...\n*****************************************************************************************
************")

if genDurations:
    type = "General"
    durations = genDurations
    dm.CreateFolder(outLocation, type)
    outputPath = outLocation + "\\General\\"
    arcpy.AddMessage("\nRunning PMP analysis for storm type: " + type)
    pmpAnalysis(basin, type, durations)         # Calls the pmpAnalysis() function to calculate the general storm PMP
    arcpy.AddMessage("\nGeneral storm analysis
complete...\n*****************************************************************************************
************")

if coolDurations:
    type = "CoolSeason"
    durations = coolDurations
    dm.CreateFolder(outLocation, type)
    outputPath = outLocation + "\\CoolSeason\\"
    arcpy.AddMessage("\nRunning PMP analysis for storm type: " + type)
    pmpAnalysis(basin, type, durations)         # Calls the pmpAnalysis() function to calculate the Cool Season storm PMP
    arcpy.AddMessage("\nCool season storm analysis
complete...\n*****************************************************************************************
************")

#if arcpy.Describe(outputTable).name:
#    outputBasAveTable()

#arcpy.RefreshTOC()
#arcpy.RefreshActiveView()
#del mxd, df
```

Appendix I

PMP Version Log: Changes to Storm Database and Adjustment Factors

PMP Versions

Version 1.0 – 1/25/2020

- Created 3 Transposition zones. Added transposition constraints to all storms
- Initial run; included GTF upper limit of 1.50 and lower limit 0.50
- MTF was set to 1 to remove from total adjustment factor.
- Precipitation frequency values for Montana, Saskatchewan, and Manitoba were not available. Used an average of the bordering NOAA Atlas 14 values for each.
- (SPAS 1336_1) Springbrook, MT – Ran as a hybrid storm as both General and Local. Used 24hr precipitation frequency values from Wyoming mini analysis for both.

<u>General Storms</u>

- SPAS 1048_1 (Hokah, MN) – Moved to zone 1
- SPAS 1206_1 (Big Rapids, MI) – Moved to zone 1
- SPAS 1297_1 (Warroad, MN) – Moved to zone 1
- SPAS 1325_1 (Savageton, WY) – Moved to zone 3
- SPAS 1335_1 (Warrick, MT) – Moved to zones 2 & 3
- SPAS 1337_1 (Parkman, SK) – NOAA Atlas 14 domain did not cover this storm center. Estimated from nearest values. Moved to zones 1 & 2
- SPAS 1433_1 (Collinsville, IL) – Moved to zone 1
- SPAS 1502_1 (Veteran, AB) – Moved to zones 2 & 3
- SPAS 1504_1 (Pelican Mountain, AB) – Moved to zone 2
- SPAS 1527_1 (Ida Grove, IA) – Moved to zone 1
- SPAS 1583_1 (Council Grove, KS) – Moved to zone 1
- SPAS 1630_1 (Bolton, ONT) – Moved to zone 1
- SPAS 1697_1 (Ironwood, MI) – Moved to zone 1
- SPAS 1738_1 (Harlan, IA) – Moved to zone 1

<u>Hybrid Storms</u>

- SPAS 1183_1 (Edgerton, MO) – Moved to zones 1 & 2
- SPAS 1228_1 (Fall River, KS) – Moved to zone 1
- SPAS 1286_1 (Aurora College, IL) – Moved to zone 1
- SPAS 1296_1 (Duluth, MN) – Moved to zones 1 & 2
- SPAS 1336_1 (Springbrook, MT) – Moved to zones 2 & 3
- SPAS 1699_1 (Hayward, WI) – Moved to zone 1
- SPAS 1725_1 (Leonard, ND) – Moved to zones 1 & 2

<u>Local Storms</u>

- SPAS 1030_1 (David City, NE) – Moved to zones 1 & 2
- SPAS 1033_1 (Ogallala, NE) – Moved to zones 2 & 3

- SPAS 1035_1 (Forest City, MN) – Moved to zone 1
- SPAS 1036_1 (Pawnee Creek, CO) – Moved to zones 2 & 3
- SPAS 1177_1 (Vanguard, SK) – Moved to zones 1, 2, & 3
- SPAS 1209_1 (Wooster, OH) – Moved to zone 1
- SPAS 1210_1 (Minneapolis, MN) – Moved to zones 1 & 2
- SPAS 1220_1 (Dubuque, IA) – Moved to zone 1
- SPAS 1226_1 (College Hill, OH) – Moved to zone 1
- SPAS 1324_1 (Glen Ullin, ND) – Moved to zones 1, 2, & 3
- SPAS 1334_1 (Buffalo Gap, SK) – Moved to zones 1, 2, & 3
- SPAS 1426_1 (Cooper, MI) – Moved to zone 1
- SPAS 1427_1 (Boyden, IA) – Moved to zone 1
- SPAS 1434_1 (Holt, MO) – Moved to zone 1
- SPAS 1673_1 (Harrow, ONT) – NOAA Atlas 14 domain did not cover this storm center. Estimated from nearest values. Moved to zone 1
- SPAS 1726_1 (Turtle River, ND) – Moved to zones 1, 2, & 3
- SPAS 1727_1 (Drummond, WI) – Moved to zone 1
- SPAS 1728_1 (Cross Plains, WI) – Moved to zone 1
- SPAS 1729_1 (Fountain, MI) – Moved to zone 1
- SPAS 1736_1 (Stanton, NE) – Moved to zones 1 & 2

Version 1.a – 1/25/2020

- Dewpoint rasters do not cover all grid points into Canada for SPAS 1210_1, 1226_1, 1673_1, & 1726_1 when calculating MTF. Extended Existing Isolines into Canada
- Calculated MTF

Version 2 – 2/12/2020

- Based off version 1

<u>Local storms</u>

- Removed SPAS 1226_1 (College Hill, OH) from local storms for sensitivity
- Added SPAS 1521_2 (Bassano, AB) to Local Storms. Moved to zones 2 & 3
- Added SPAS 1734_1 (Thief River Falls, MN) to local storms. Moved to zones 1, 2, &3

<u>Hybrid storms</u>

- Limited SPAS 1336_1 (Springbrook, MT) to west of 99° W or greater than 1,500 ft elevation

<u>General storms</u>

- Added SPAS 1735_1 (Coldwater, MI) to general storms. Moved to zones 1, 2, & 3

- Limited SPAS 1325_1 (Savageton, WY) to west of the Dakotas and south of 46° N
- Limited SPAS 1206_1 (Big Rapids, MI) to east of 99° W

Cool Season storms

- Added cool season storms. One storm type using 24-hr 100-yr precipitation frequency values. Moved all storms to all zones.
- Added SPAS 1245_1 (Ashland, WI)
- Added SPAS 1698_1 (Bellefontaine, OH)
- Added SPAS 1732_1 (Madison, SD)
- Added SPAS 1733_1 (Groton, SD)
- Added SPAS 1737_1 (Chan Gurney, SD)
- Added SPAS 1739_1 (Iron River, MI)

Version 3 – 3/3/2020

This version used everything that was done in version 2 with the following changes

Cool Season storms

- Added SPAS 1740_1 (Croswell, MI)
- Added SPAS 1743_1 (Belcourt, ND)

Local storms

- Added SPAS 1744_1 (East Trout Lake, SK) to local storms. Moved to zones 1, 2, & 3
- SPAS 1177_1 (Vanguard, SK) - Updated storm center 100-yr precipitation frequency value from 2.80 to 2.70 as part of Frenchman PMP PF update
- SPAS 1334_1 (Buffalo Gap, SK) - Updated storm center 100-yr precipitation frequency value from 3.32 to 2.80 as part of Frenchman PMP PF update

General storms

- SPAS 1335_1 (Warrick, MT) - Updated storm center 100-yr precipitation frequency value from 4.82 to 4.09 as part of Frenchman PMP PF update
- SPAS 1502_1 (Veteran, AB) - Updated storm center 100-yr precipitation frequency value from 3.14 to 3.61 as part of Frenchman PMP PF update
- SPAS 1325_1 (Savageton, WY) – Limited GTF to 1. All grid cells had a GTF value above 1. IMPF increases storm by 19% this is only adjustment applied.
- SPAS 1337_1 (Parkman, SK) – Increased the lower limit of the GTF from 0.50 to 1.15

Hybrid storms

- SPAS 1336_1 (Springbrook, MT) - Updated storm center 100-yr precipitation frequency value from 4.56 to 2.79 for Local and 3.52 for General as part of Frenchman PMP PF update. Updated transposition limits from West of 99° W or greater than 1,500 ft to all of zones 2 and 3.

- SPAS 1699_1 (Hayward, WI) – Increased the lower limit of the GTF from 0.50 to 0.75.

Version 3a – 3/9/2020

Used version 3 with these changes

Hybrid storms (Local)

- SPAS 1336_1 (Springbrook, MT) – Removed from Local storms

Hybrid storms (General)

- SPAS 1336_1 (Springbrook, MT) – Updated transposition restraint to west of 98° W and set the GTF to 1. There were no GTF values below 1 before this was applied.

General storms

- SPAS 1337_1 (Parkman, SK) – Removed GTF constraint applied in v3

Version 3b – 3/9/2020

Used version 3a with these changes

General storms

- SPAS 1335_1 (Warrick, MT) – Updated transposition restraint to west of 98° W and set GTF to 1.

Version 3c – 3/9/2020

Used version 3b with these changes

General storms

- SPAS 1335_1 (Warrick, MT) –Set GTF to 1.1.

Hybrid storms (General)

- SPAS 1336_1 (Springbrook, MT) – Set the GTF to 1.1.

Version 3d – 3/10/2020

Used version 3b with these changes

General storms

- SPAS 1335_1 (Warrick, MT) –Normalized the GTF to a maximum value of 1.

Hybrid storms (General)

- SPAS 1336_1 (Springbrook, MT) – Normalized the GTF to a maximum value of 1.

Version 3b-Check overlap with WY-compare to adjacent PMP studies (Quad Cities), check against HMR 51 and NOAA Atlas 14

Version 4 – 3/12/2020

Used version 3 with these changes

Hybrid storms

- SPAS 1336_1 (Springbrook, MT) – Removed from Local storms
- SPAS 1336_1 (Springbrook, MT) – Updated transposition restraint to west of 98° W. Updated storm center 100-yr precipitation frequency value back to 4.56 from 3.52. GTF values were unrealistic.

General storms

- SPAS 1337_1 (Parkman, SK) – Removed GTF constraint applied in v3
- SPAS 1335_1 (Warrick, MT) - Updated transposition restraint to west of 98° W. Updated storm center 100-yr precipitation frequency value back to 4.82 from 4.09. GTF values were unrealistic
- SPAS 1502_1 (Veteran, AB) - Updated storm center 100-yr precipitation frequency value back to 3.14 from 3.61.

Local storms

- SPAS 1177_1 (Vanguard, SK) - Updated storm center 100-yr precipitation frequency value back to 2.80 from 2.70. GTF values were unrealistic
- SPAS 1334_1 (Buffalo Gap, SK) - Updated storm center 100-yr precipitation frequency value back to 3.32 from 2.80

Version 4a – 3/22/2020

Used version 4 with these changes

Local storms

- SPAS 1177_1 (Vanguard, SK) - Normalized GTF values to a maximum of 1.4. This reduced GTF values by 7%
- SPAS 1334_1 (Buffalo Gap, SK) – Normalized GTF values to a maximum of 1.2. This reduced GTF values by 20%

General storms

- SPAS 1335_1 (Warrick, MT) – Updated transposition restraint to zones 2 and 3 and limited GTF factor to a maximum of 1
- SPAS 1336_1 (Springbrook, MT) – Updated transposition restraint to zones 2 and 3 and limited GTF factor to a maximum of 1

Version 4b – 3/24/2020
Used version 4a with these changes
Local storms

- SPAS 1744_1 – East Trout Lake, SK – Removed from list

Version 4c – 5/1/2020
Used version 4b with these changes
Local storms

- SPAS 1177_1 – Vanguard, SK – Normalized GTF values to a maximum of 1.2. This reduced GTF values by 20%

Version 5 – 7/9/2020
Used version 4c with these changes

- Updated precipitation frequency values. NOAA atlas 14 precipitation frequency values were unavailable for areas in Canada and Montana. A mini analysis was completed for those areas and merged with existing NOAA atlas values at the borders.

Local storms

- SPAS 1177_1 – Vanguard, SK – Updated storm center precipitation from 2.80 to 2.86. Normalized GTF values to a maximum of 1.2. Limited to areas above 1,400 ft.
- SPAS 1744_1 – East Trout Lake, SK – Added back to list. Normalized GTF to a maximum of 1.2. Limited to zones 2 & 3 above 48° N

General Storms

- SPAS 1335_1 – Warrick, MT – Updated storm center precipitation amount from 4.82 to 5.00.

Version 5 Notes
6hr General storms: How to handle the boundary between 2 and 3? Move 1336 further east? What is the actual difference in values? What adjustments have we applied to 1336 for v5?
72hrs how we lower Savageton more?
Local storm-1744, let it go into zone 2 a little?

Version 5a – 7/11/2020
Used version 5 with these changes

General Storms

- SPAS 1336_1 (Springbrook, MT) – Updated transposition constraints from zones 2 and 3 to zones 1,2 & 3 but above 1,400 ft.
- SPAS 1325_1 (Savageton, WY) – Updated transposition constrain to anything above 4,000 ft

Local Storms

- SPAS 1744_1 (East Trout Lake, SK) – Updated transposition constraints from zones 2 & 3 above 48° N to zones 1,2,3 above 47° N and west of 101°W

Version 5a notes:
General storm, 6hr 1000sqmi, may need to move 1336 to all of zone 1 to get perfect fit-not sure if it matters at 6hr 1000
Local storm 6hr and 24hr 100sqmi-what does it look like without 1744? Can we use elevation constraints instead of lat/lon to fit better?

Version 5b – 7/12/2020
Used version 5a with these changes
General Storms

- SPAS 1336_1 (Springbrook, MT) – Updated transposition constraints from zones 1,2 & 3 above 1,400 ft to all zones.

Local Storms

- SPAS 1744_1 (East Trout Lake, SK) – Removed from storm list

Version 5c – 7/12/2020
Used version 5a with these changes
Local Storms

- SPAS 1744_1 (East Trout Lake, SK) – Updated transposition to above 1,500 ft and below 2,500ft elevation and west of 101°N

Version 5d – 1/18/2020
Used v5c with these changes
General Storms

- SPAS 1206_1 – Updated transposition to go to all of zone 1

Version 5e – 1/21/2021
Used v5c with these changes
General Storms

- SPAS 1206_1 – Updated transposition to go to all of zone 1 and 2 East of 103°
- SPAS 1502_1 – Updated transposition to go to all 3 zones

Local Storms

- SPAS 1744_1 – Normalized the GTF down to a maximum of 1.05 down from 1.2 to try to get it to better fit other storms at larger area sizes.

Version 5f – 2/1/2021
Used v5e with these changes
Local Storms

- SPAS 1744_1 – Normalized the GTF down to a maximum of 1 and allowed to be transposed to all zones.

Need comparisons between storm types and for all HMR 51 area sizes
Remove 1744 again or cap it so it basically has no influence-then what happens
Check what is happening in circled areas
Why the transition from 1177 to 1324 at 6hr 100sqmi in zone 1 (elevation?)
Move 1286 further west? Or what if only used as a general storm?

Version 5g – 2/5/2021
Used v5f with these changes – **This is final version**
Local Storms

- SPAS 1286_1 – Updated transposition to allow storm to go to all zones
- SPAS 1699_1 – Updated transposition to allow storm to go to all zones

Appendix J
Snow Water Equivalent and Temperature Time Series

1.1 Overview

Applied Weather Associates (AWA) developed gridded 100-year snowpack in conjunction with an average daily temperature timeseries. The information was developed to cover a timeframe representing a complete picture of snow accumulation and snowmelt throughout the region. Therefore, it is important to note that the meteorological conditions associated with the full Probable Maximum Precipitation (PMP) rainfall event are valid from June through September over the North Dakota region. Therefore, no direct snowmelt is expected to occur during the PMP rainfall event. This is consistent with all PMP studies in the region completed by AWA (Tomlinson et al., 2008; Kappel et al., 2014; Kappel et al., 2018) and with Hydrometeorological Reports 51 and 55A (Schreiner and Riedel, 1978 and Hansen et al. 1988).

1.2 Development of Meteorological Time Series

Snowmelt calculations are dependent on the availability of reliable snowpack and temperature climatologies. For this study, several gridded data sources along with point location (surface station data), were evaluated and used to develop the gridded data sets used for the analysis as described below.

1.2.1 Snow Water Equivalent

AWA utilized surface observations, remote sensing data and modeled gridded data to quantify the spatial and temporal 100-year snow water equivalent (SWE) values.

Station Data

AWA calculated the 100-year (1% Exceedance) point value SWE based on 194 surface stations from Snow Telemetry (SNOTEL) and Global Historical Climatology Network (GHCN) data networks within and surrounding the basins. SWE data were extracted for the 1st and the 15th of each month from March 1st through September 15th. For each date, for each station the 100-year SWE was calculated based on station L-moments statistics and the generalized extreme value (GEV) probability distribution (Hosking and Wallis, 1997). The GHCN network sometimes provided direct measurements of SWE but always provided direct measurements of snow depth. An average snow density of 25% was applied for each date to convert the GHCN snow depth data to SWE (Pomeroy and Gray, 1995).

SNODAS Data

In addition to the point snow water equivalent, AWA utilized the National Operational Hydrologic Remote Sensing Center (NOHRSC) SNOw Data Assimilation System (SNODAS) gridded dataset. SNODAS integrates observed, remotely sensed, and modeled datasets into estimated snowpack variables. SNODAS is a physically-based, near real-time energy and mass balance, spatially-uncoupled, vertically-distributed, multi-layer snow model (Carroll et al., 2001; NOHRSC, 2004). The model has high spatial (1-km^2) and temporal (1-hour and daily) resolutions and is run for the conterminous United States. Snowpack products generated by SNODAS include SWE, snow depth, snowpack average temperature, snowmelt, and surface and blowing snow sublimation. The SNODAS data are available starting in 2003 though present, providing 16 years of data to create snowpack climatologies for the lower 48 states. Since portions of the North Dakota domain extend into Canada, the unclipped SNODAS data were need and available starting in 2013 though present, providing 7 for this study.

SNODAS daily gridded data were utilized on the 1st and the 15th of each month, starting March 1st through September 15th. These data were utilized to calculate the 1% exceedance (100-year), mean, maximum, and minimum snowpack spatial variation. The SNODAS dataset were used to derive snowpack 1% Exceedance (1% Exceedance = average + (2.3263 * St. Dev)) spatial snowpack climatologies. 2.3263 is the z-score (standard deviation) of the 99th percentile from the normal distribution (R example qnorm(0.99) = 2.326348). The daily gridded SNODAS climatologies were used to aid in the spatial interpolation of the station 100-year SWE (i.e., scaling the gridded spatial pattern to the observed station 100-year SWE). The final adjusted 100-year SWE grids were derived using the estimated SNODAS 100-year climatologies and station data 100-year SWE estimates as input into a climatologically aided spatial interpolation process following Daly *et al.*, (1994), Schaake *et al.,* (2004), Hultstrand and Kappel (2017), and Hultstrand and Fassnacht (2020).

Daymet Data

In addition to the SNODAS and station data, AWA utilized NASA's Oak Ridge National Laboratory Daymet gridded dataset. Daymet is a collection of gridded estimates of daily weather parameters generated by interpolation and extrapolation from daily meteorological observations (Thornton et al., 2016). The model has high spatial (1-km^2) and temporal (daily) resolutions and is run for all of North America. Daymet products include SWE and are available starting in 1980 and though present, providing 40 years of data to help create snowpack climatologies.

Like the other data sets, AWA utilized daily gridded Daymet data for the 1st and the 15th of each month, starting March 1st and continuing through September 15th. These data were used to calculate the 1% Exceedance (100-year), mean, maximum, and minimum snowpack spatial variation. The Daymet dataset was used to derive snowpack 1% Exceedance (1% Exceedance = average + (2.3263 * St. Dev)) spatial snowpack climatologies. The daily gridded Daymet climatologies were used to aid in the spatial interpolation of the station 100-year SWE (i.e., scaling the gridded spatial pattern to the observed station 100-year SWE). The final adjusted 100-year SWE grids were derived using the estimated SNODAS 100-year climatologies and station data 100-year SWE estimates as input into a climatologically aided spatial interpolation process following Daly *et al*., (1994), Schaake *et al.,* (2004), Hultstrand and Kappel (2017), and Hultstrand and Fassnacht (2020).

Final 100-year SWE Climatologies

Comparison of the station observed 100-year SWE estimates to the gridded 1% exceedance estimates for SNODAS and Daymet were made for the 1st and the 15th of each month from March 1st through September 15th. The SNODAS and Daymet 1% exceedance estimates had similar goodness-of-fit measurements (mean error (ME), mean absolute error (MAE), root mean square error (RMSE), correlation coefficient (r)). Based on the goodness-of-fit measurements, and available period of record (POR) the Daymet grids were adjusted to the surface station 100-year estimates and the SNODAS data were not used in the final grid development.

The final adjusted 100-year gridded SWE climatologies were produced from a combination of station data and SNODAS for the 1st and the 15th of each month from March 1st through September 15th. The gridded SWE values utilized for snowmelt calculations over the drainage basins are illustrated below for March 1st, March 15th, April 1st, and April 15th, May 1st, May 15th, June 1st, and June 15th (**Error! Reference source not found.** through Figure 8).

1.2.2 Temperature Time Series

Gridded daily surface average temperature timeseries (T_a) were compiled and utilized as input for the gridded snowmelt calculations as described in the following snowmelt methodology section (Section 0). Daily gridded temperature data was obtained from the Parameter-elevation Regressions on Independent Slopes Model (PRISM) (PRISM, 2004). The Daymet temperature products were also obtained and considered as an alternative option for snowmelt and sensitivity analysis of the snowmelt model but were not used in the final snowmelt tool calculations. The daily minimum (T_{a_min}), daily maximum (T_{a_max}), and daily average (T_{a_avg}) parameters were each evaluated. These data sets covered the same time period as the SWE 100-year climatology but were developed on a daily timestep. The gridded T_{a_avg} Daymet values utilized for snowmelt calculations over the drainage basins are illustrated below for March 1st, March 15th, April 1st, and April 15th, May 1st, May 15th, June 1st, and June 15th (**Error! Reference source not found.** through Figure 16).

1.3 Methodology

1.3.1 Snowmelt Equation

The U.S. Army Corps of Engineers (USACE) has conducted numerous snowmelt studies, which were aimed primarily at providing procedures for deriving maximum snow melt design floods. The USACE summarized two approaches to compute snowmelt. The first is the energy budget method, which allows the snowmelt solution to be as physically based as practicable by incorporating into snowmelt equations factors such as solar radiation, wind, and long-wave radiation exchange. The second method, temperature index equation, is a more simplified approach in which air temperature is assumed to be a representative index of all energy sources so that it can be used as the sole independent variable in calculating snowmelt. The energy budget equation for a rain-free situation with a forested area of 60-80% is defined as:

$$M = k(0.0084v)(0.22T_a + 0.78T_d) + 0.029T_a \qquad Eq. 1$$

where M is snowmelt (inches/day), T_a is air temperature (°F), T_d is dew point temperature (°F), and v is the wind speed (mph).

The Temperature Index equation is defined as:

$$M = C_m(T_a - T_b) \qquad Eq. 2$$

where M is snowmelt (inches/day), C_m is melt rate coefficient, T_a is the air temperature (^oF), and T_b is the base air temperature of 32.0^{o}F. The range of the C_m factor is typically between 0.04 and 0.08 inches/ ^{o}F for rain-free situations and up to 0.18 for rain-on-snow situations. A C_m factor of 0.06 is a common factor used when other calibration and/or snowmelt information is limited and for generally rain free snowmelt scenarios (USACE, 1998). For this study, four C_m factors are available (Table 1).

Table 1. Temperature index model melt rate coefficient used in PMP tool.

Melt Factor (Cm)	General Description
0.060	Clear sky, limited wind, the melt factor for clear day can range between 0.04 and 0.08
0.187	Heavy Rain, 10mph Wind, 10" per 24hr period
0.270	Heavy Rain, 20mph Wind, 10" per 24hr period
0.353	Heavy Rain, 30mph Wind, 10" per 24hr period

1.4 GIS Data Preparation and Snowmelt Tool

Geographic Information Systems (GIS) were utilized to facilitate spatial data management, spatial analysis, and mapping. Temperature and SWE gridded datasets were obtained and processed in the Network Common Data Form (netCDF) multidimensional file format. Climate Data Operators (CDO) (Schulzweida, 2019) were used to process gridded files in netCDF format. Processing gridded files involved calculating daily ensemble means for the available period of record for the T_a datasets, resampling the T_a and SWE gridded datasets to the 90 arc-second spatial resolution grid format, using the WGS 1984 coordinate system, and converting the gridded datasets from netCDF format to ESRI geodatabase raster format. The 90 arc-second grid network matches the grid network utilized in the North Dakota Statewide PMP Study and provides full coverage over the analysis domain at a spatial resolution sufficient to capture variations over the spatial field.

A basin snowmelt calculation tool, utilizing SWE and daily temperature derived during this study, was developed within the ArcGIS environment which allowed snowmelt calculations to be made efficiently with a variety of input parameters. Starting date, ending date, daily T_a time series gridded datasets, starting day SWE gridded dataset, degree day coefficient, and the drainage basin are all variable input parameters.

Based on the input parameters, the GIS snowmelt tool calculates the basin average snowmelt, using the Temperature Index method described in Section 0, along with the basin average values for daily T_a, degree days, and SWE. The tool also can provide the output for a discrete point location. The output is provided in a table in both ArcGIS geodatabase and Excel format.

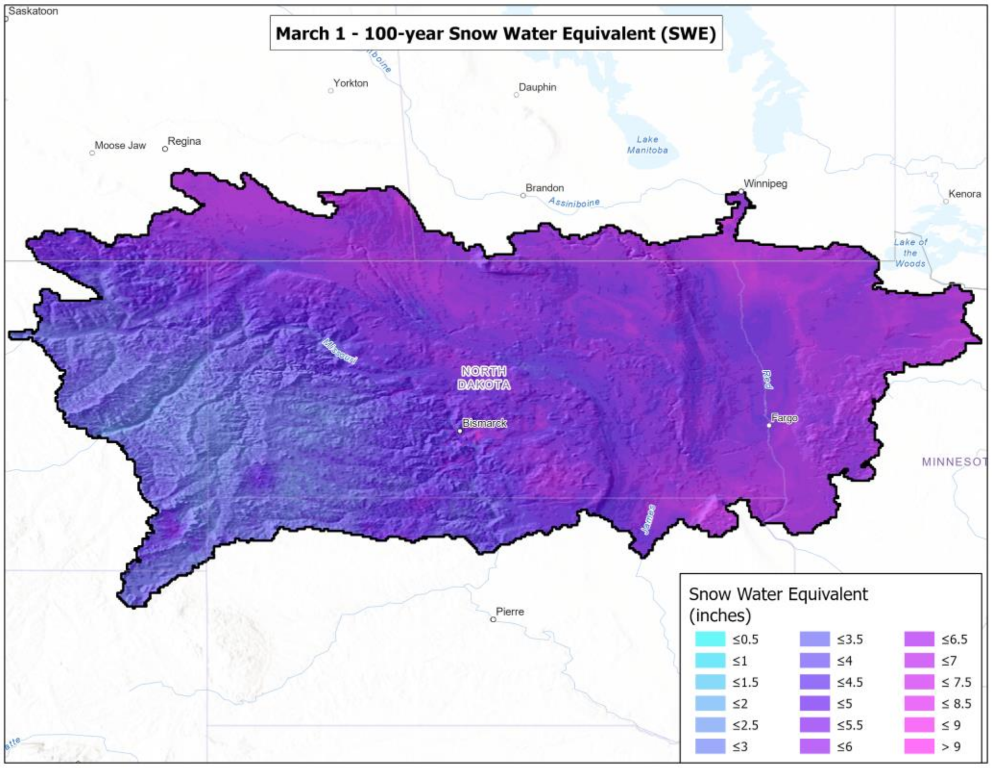


Figure 1. Final 100-year SWE for March 1.

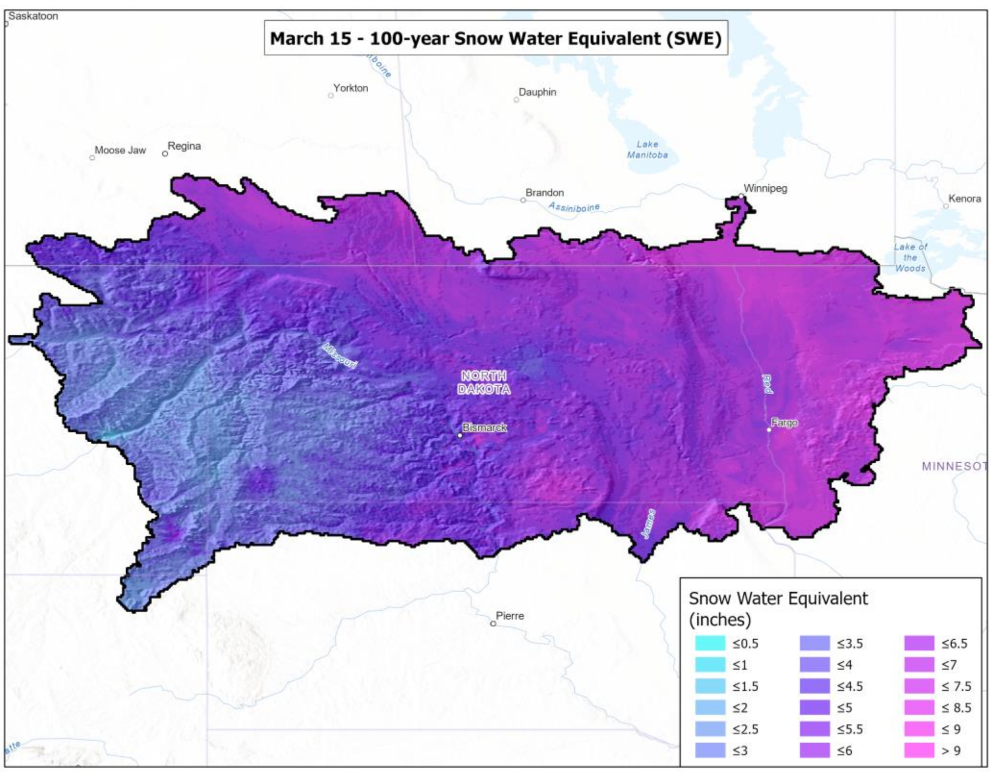


Figure 2. Final 100-year SWE for March 15.

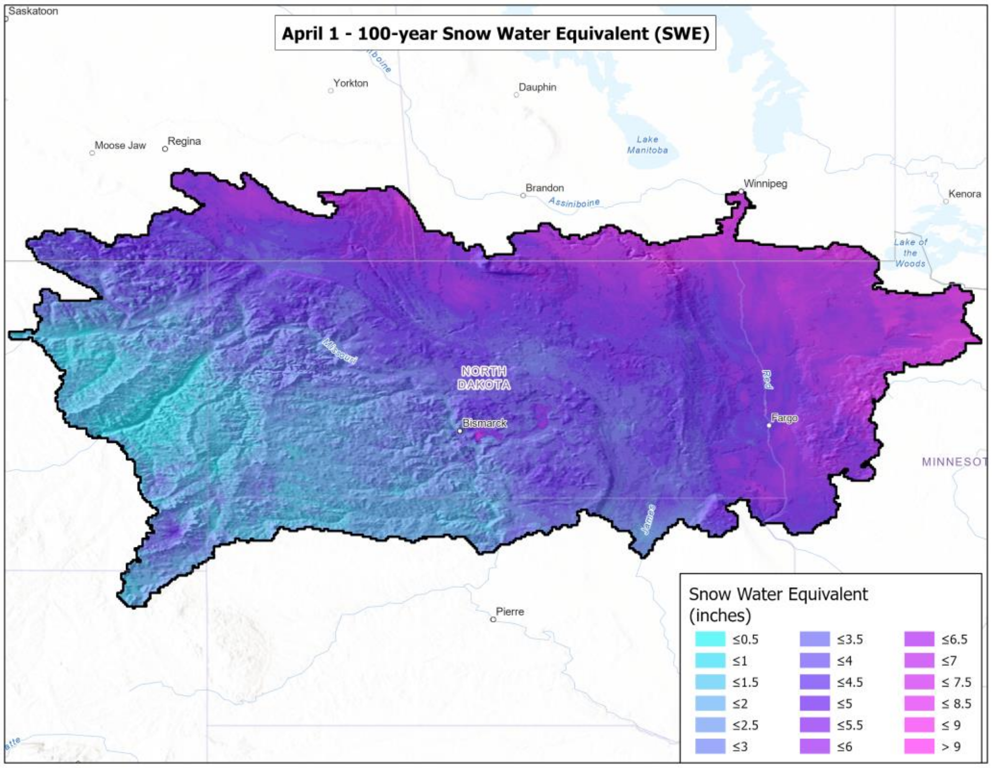


Figure 3. Final 100-year SWE for April 1.

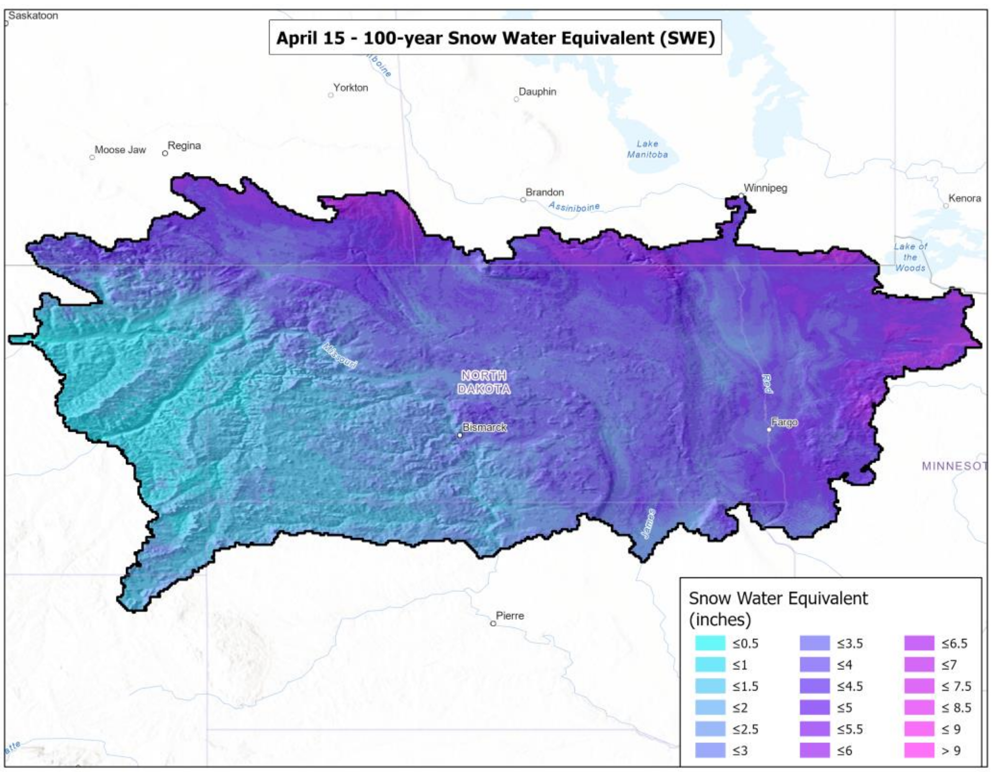


Figure 4. Final 100-year SWE for April 15.

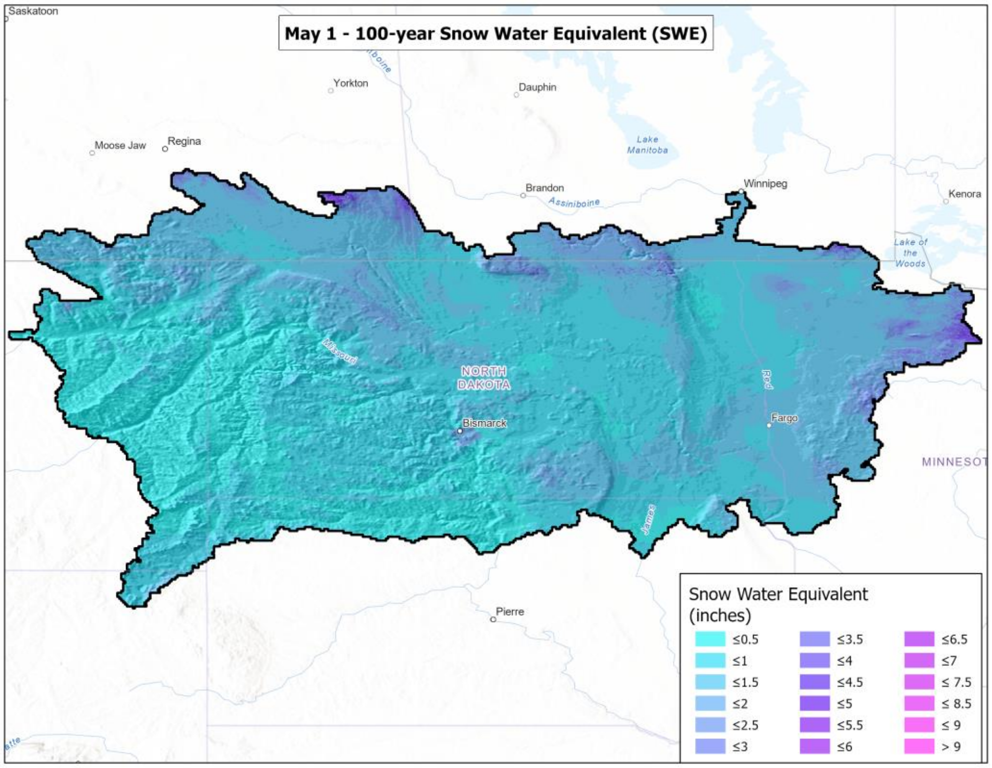


Figure 5. Final 100-year SWE for May 1.

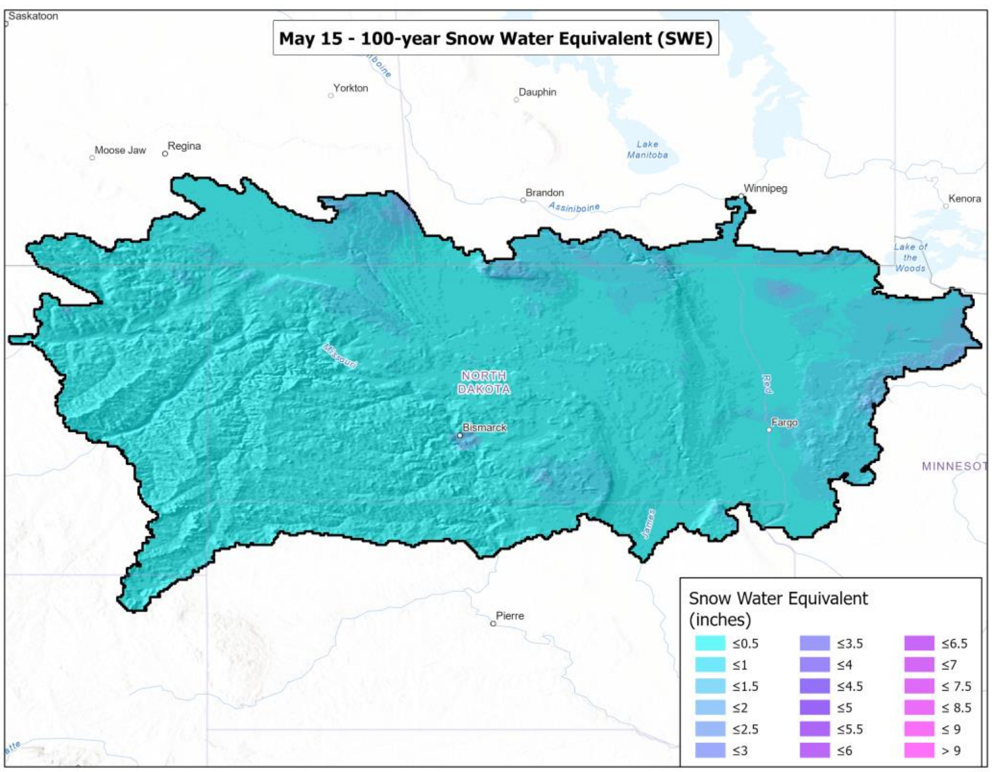


Figure 6. Final 100-year SWE for May 15.

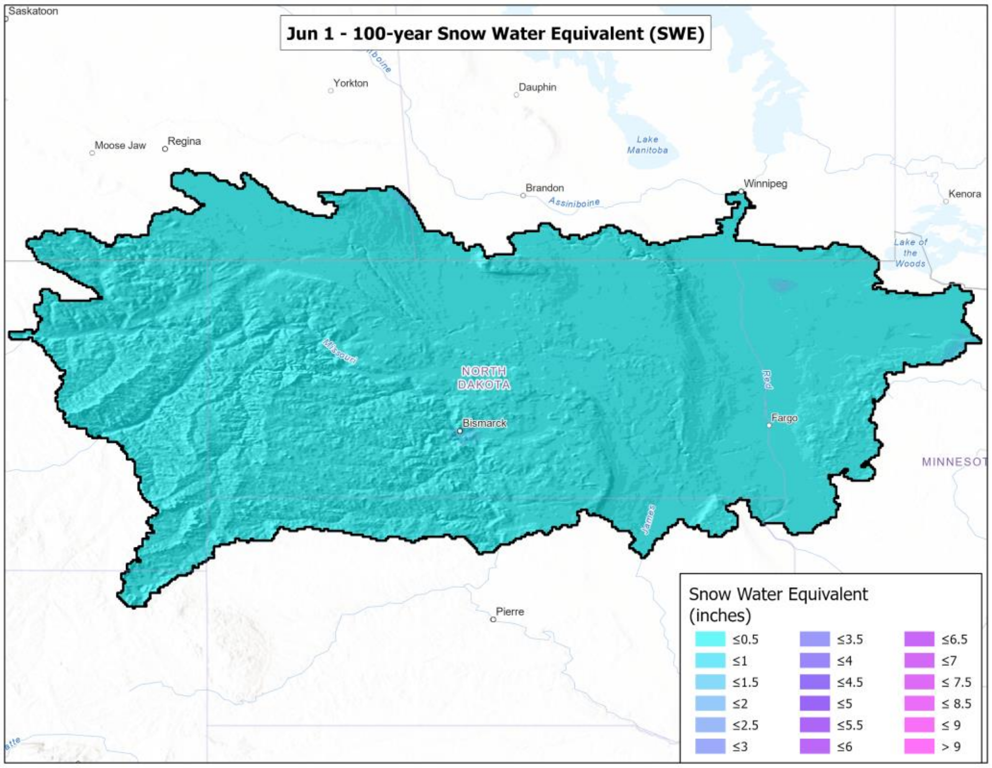


Figure 7. Final 100-year SWE for June 1.

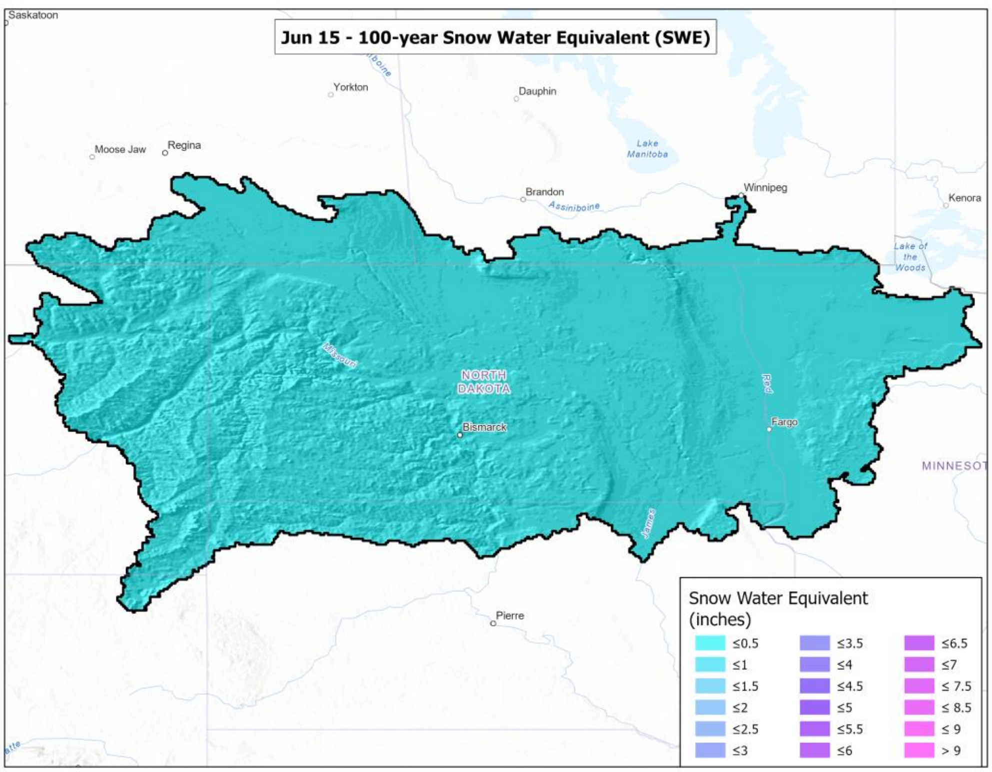


Figure 8. Final 100-year SWE for June 15

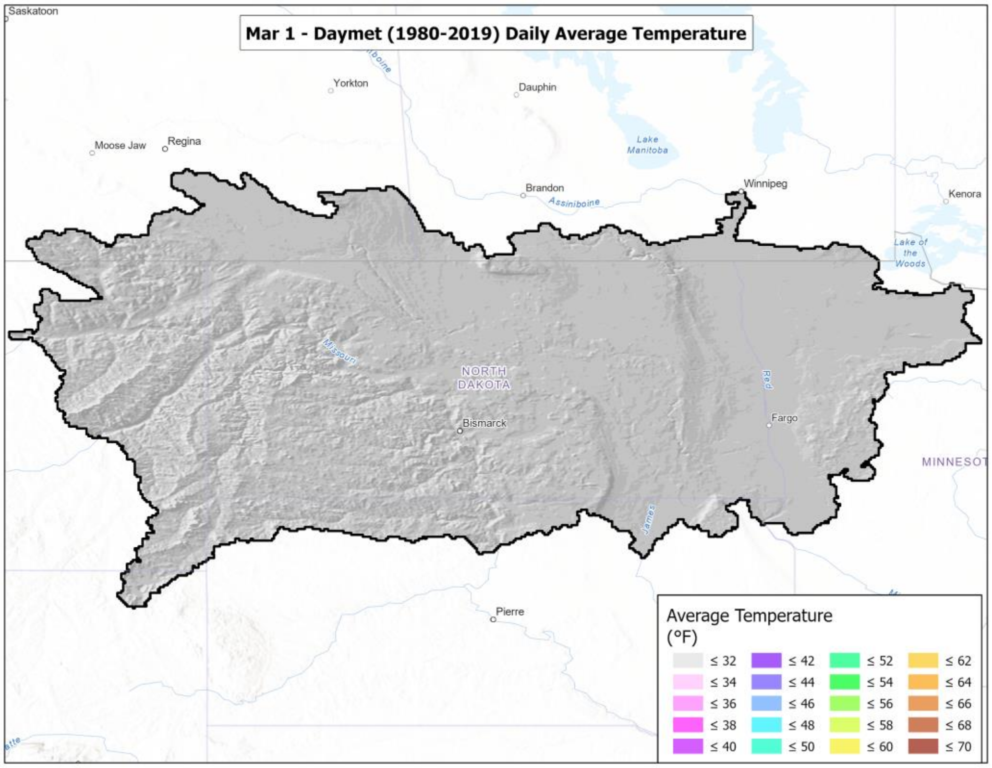


Figure 9. Daily Average Temperature for March 1.

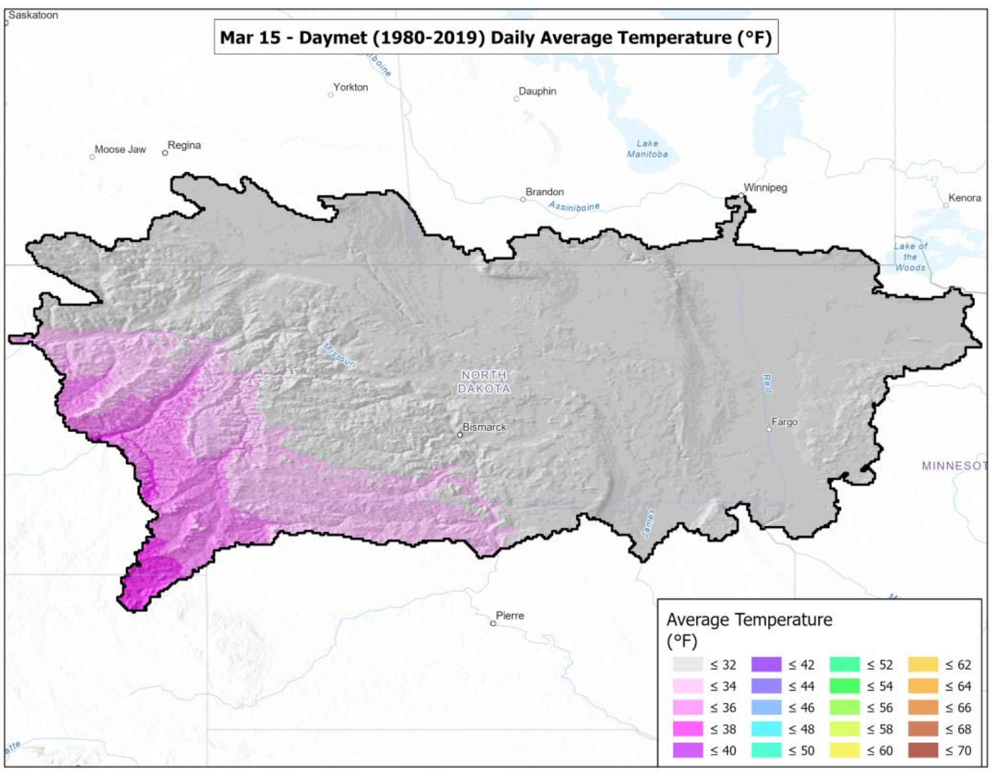


Figure 10. Daily Average Temperature for March 15.

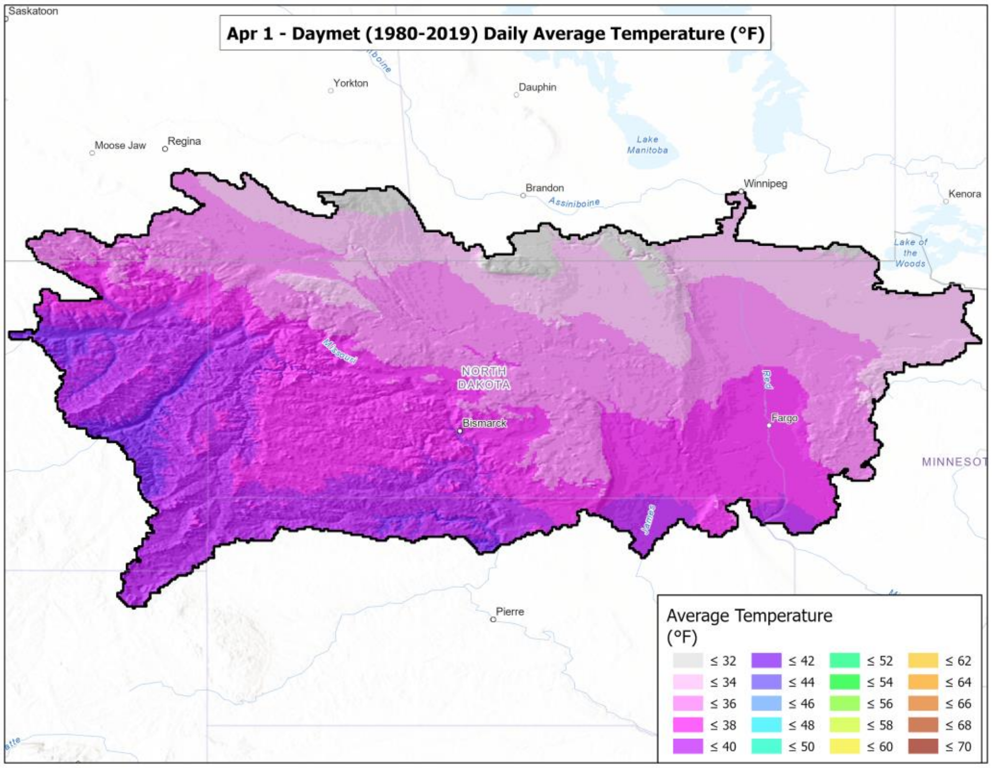


Figure 2. Daily Average Temperature for April 1.

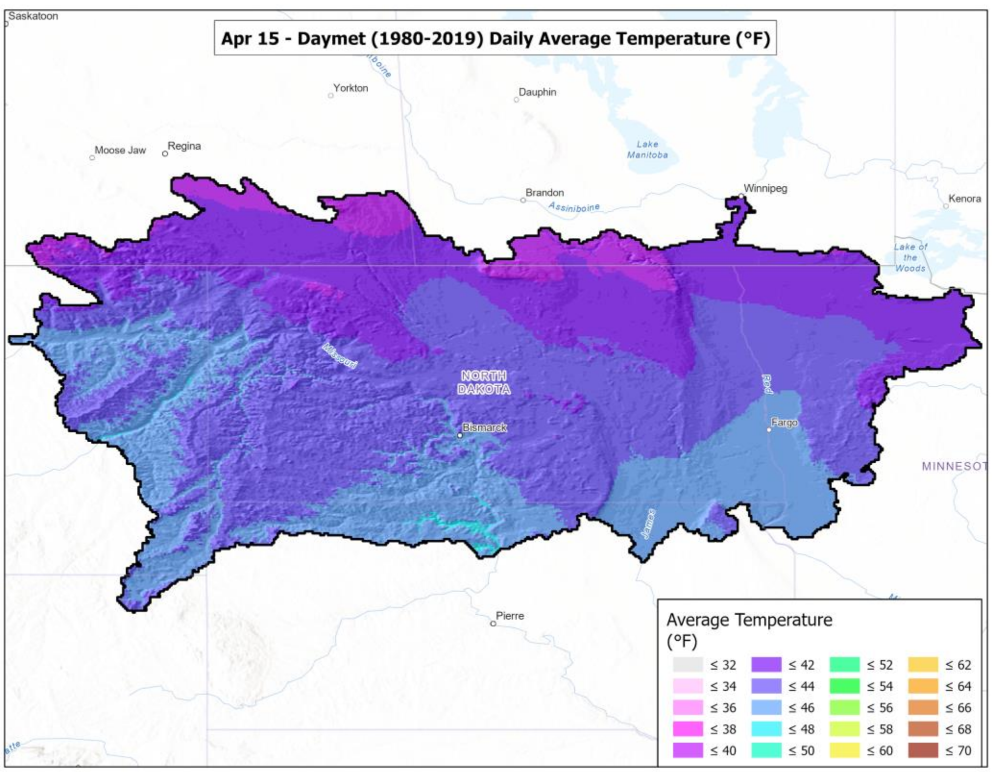


Figure 3. Daily Average Temperature for April 15.

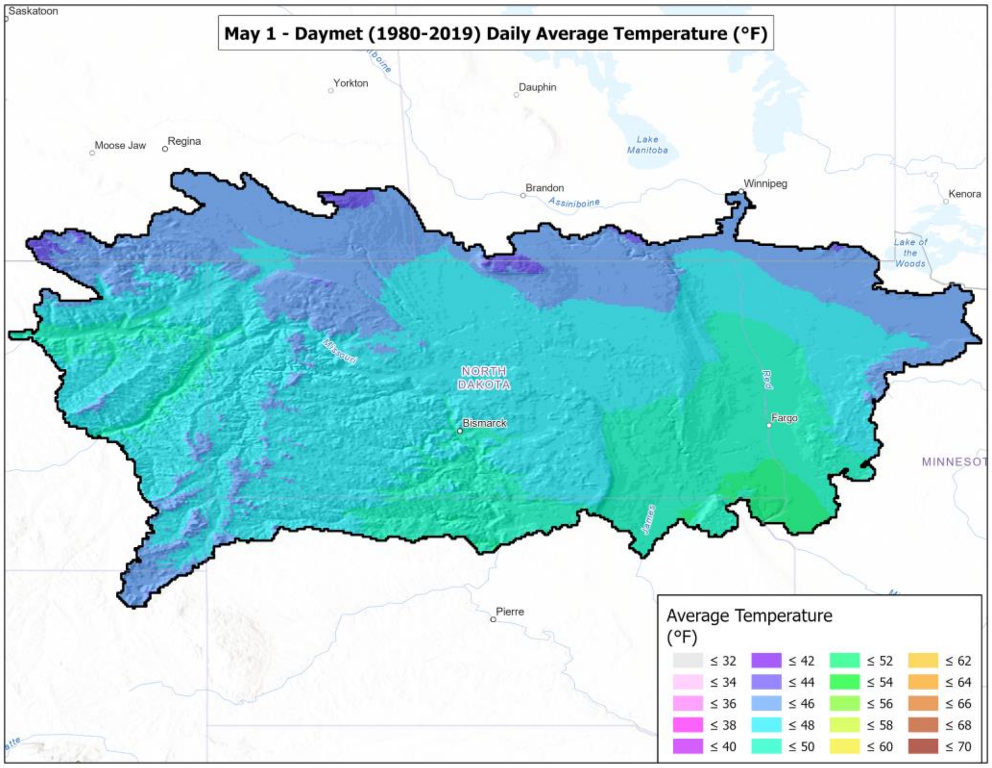


Figure 4. Daily Average Temperature for May 1.

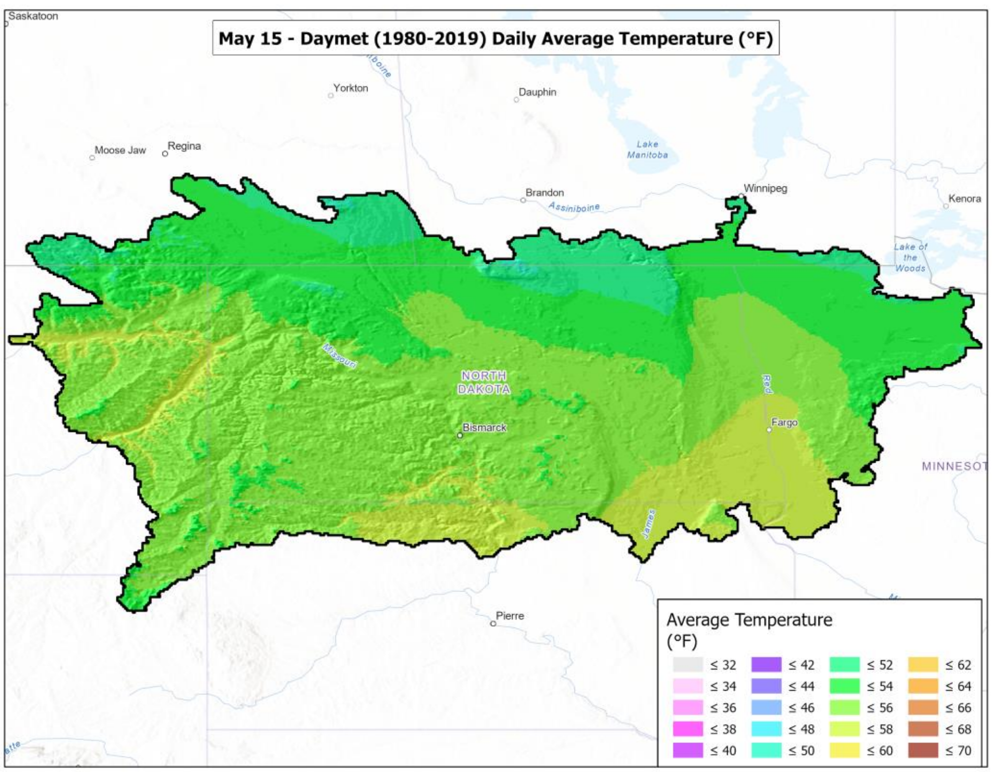


Figure 14. Daily Average Temperature for May 15.

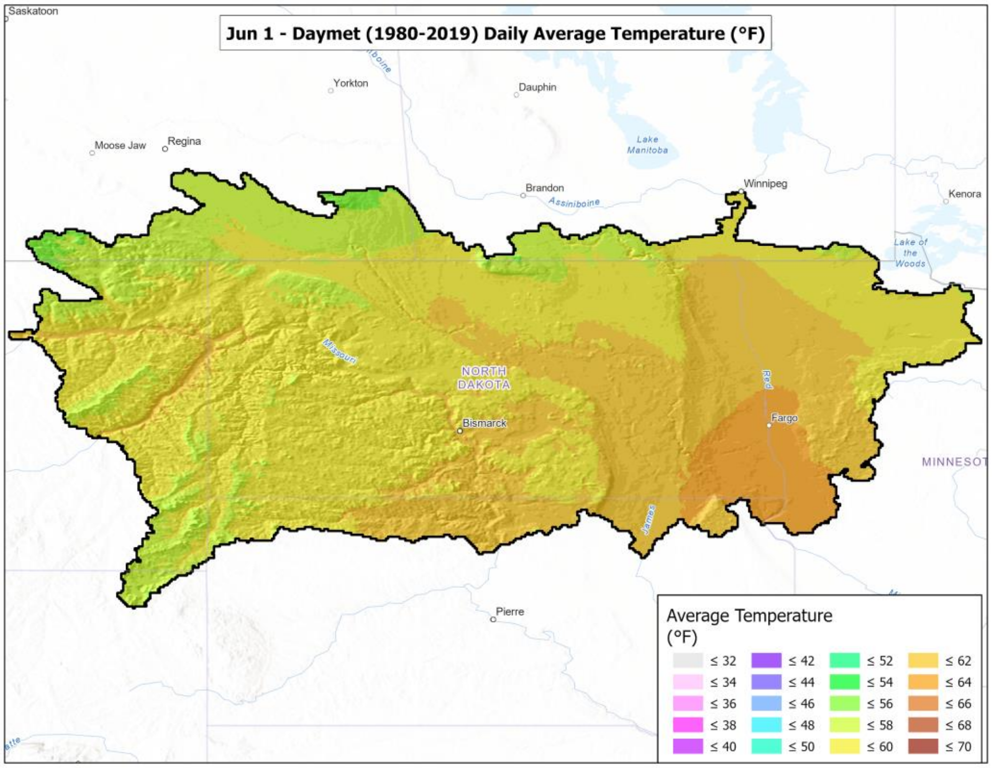


Figure 5. Daily Average Temperature for June 1.

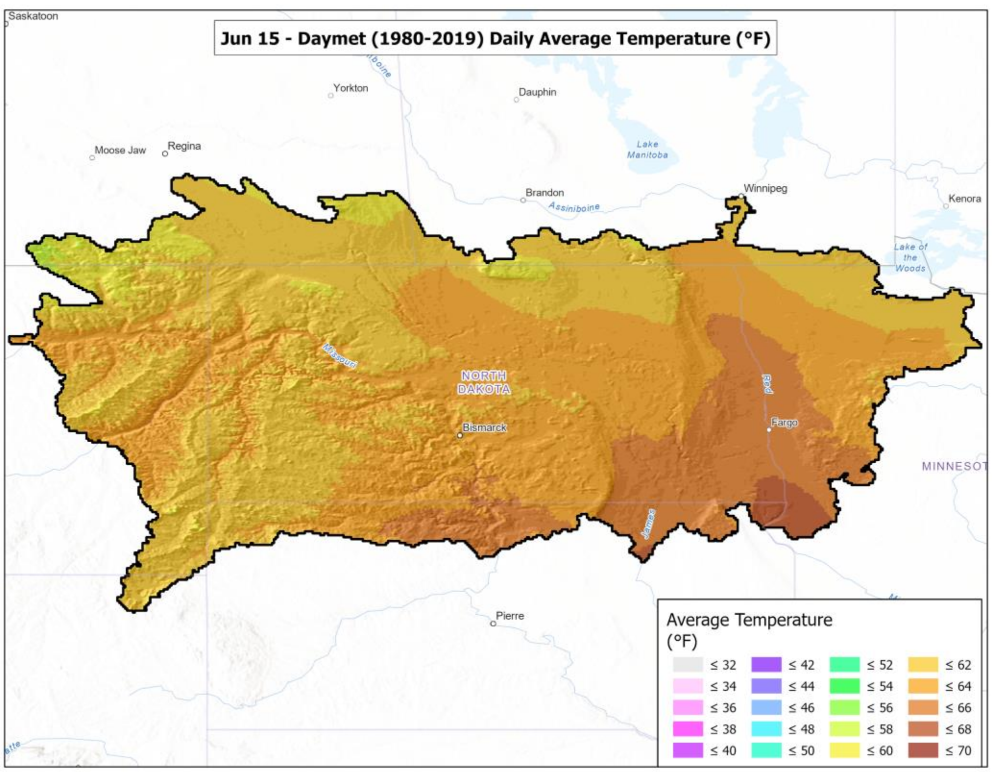


Figure 16. Daily Average Temperature for June 15.

References

Carroll, T., D. Cline, G. Fall, A. Nilsson, L. Li, and A. Rost. 2001. NOHRSC Operations and the Simulation of Snow Cover Properties for the Conterminous U.S. Proceedings of the 69th Annual Meeting of the Western Snow Conference, pp. 1-14.

Daly, C., Neilson, R.P., Phillips, D.L. 1994. A statistical-topographic model for mapping climatological precipitation for mountainous terrain. *Journal of Applied Meteorology* 33, 140-158.

Hansen, E.M., Fenn, D.D., Schreiner, L.C., Stodt, R.W., and J.F., Miller, 1988: Probable Maximum Precipitation Estimates, United States between the Continental Divide and the 103rd Meridian, *Hydrometeorological Report Number 55A*, National weather Service, National Oceanic and Atmospheric Association, U.S. Dept of Commerce, Silver Spring, MD, 242 pp.
Hosking, J.R.M. and J.R. Wallis, 1997. Regional Frequency Analysis an Approach Based on L-Moments, Cambridge University Press, 224 pp.

Hultstrand, D.M., and Kappel, W.D. 2017. The Storm Precipitation Analysis System (SPAS) Report. *Nuclear Regulatory Commission (NRC)* Inspection Report No 99901474/2016-201, Enercon Services, Inc., 95pp.

Hultstrand, D.M., and Fassnacht, S.R. 2020 (*in review*). The Best Precipitation Estimates for a Hydrologic Model by Combining Gauge and Radar Data. *Geographical Research Letters*.
Kappel, W.D., Hultstrand, D.M., Muhlestein, G.A., Steinhilber, K., McGlone, D., Parzybok, T.W, and E.M. Tomlinson, December 2014: Statewide Probable Maximum Precipitation (PMP) Study for Wyoming.

Kappel, W.D., Hultstrand, D.M., Muhlestein, G.A., Steinhilber, K., McGlone, D., Rodel, J., and B. Lawrence, November 2018: Regional Probable Maximum Precipitation for the States of Colorado and New Mexico. Prepared for the Colorado Division of Water Resources and the New Mexico State Engineers Office.

National Operational Hydrologic Remote Sensing Center. 2004. Snow Data Assimilation System (SNODAS) Data Products at NSIDC, Version 1. [Indicate subset used]. Boulder, Colorado USA. NSIDC: National Snow and Ice Data Center. doi: https://doi.org/10.7265/N5TB14TC. [Date Accessed]

Schreiner, L.C., and J.T. Riedel, 1978: Probable Maximum Precipitation Estimates, United States East of the 105th Meridian. *Hydrometeorological Report No. 51*, National weather Service, National Oceanic and Atmospheric Association, U.S. Dept of Commerce, Silver Spring, MD, 87pp.

Tomlinson, E.M., Kappel W.D., Parzybok, T.W., Hultstrand, D., Muhlestein, G., and B. Rappolt, May 2008: Site-Specific Probable Maximum Precipitation (PMP) Study for the Wanahoo Drainage Basin, Prepared for Olsson Associates, Omaha, Nebraska.

Thornton, P.E., M.M. Thornton, B.W. Mayer, Y. Wei, R. Devarakonda, R.S. Vose, and R.B. Cook. 2016. Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 3. ORNL DAAC, Oak Ridge, Tennessee, USA. https://doi.org/10.3334/ORNLDAAC/1328
Pomeroy, J.W., D.M. Gray, 1995. Snowcover: accumulation, relocation, and management. Saskatoon, Sask., Canada: National Hydrology Research Institute (NHRI).

PRISM Climate Group, Oregon State University, http://prism.oregonstate.edu, created 4 Feb 2004

Schaake, J., Henkel, A., Cong, S. 2004. Application of PRISM Climatologies for Hydrologic Modeling and Forecasting in the Western U.S. In: *18th Conference on Hydrology*, American Meteorological Society, Seattle, WA.

Schulzweida, Uwe. (2019, October 31). CDO User Guide (Version 1.9.8). http://doi.org/10.5281/zenodo.3539275

U.S. Army Corps of Engineers, 1998. Engineer Manual 1110-2-1406 - Runoff from Snowmelt. Washington DC.

Appendix K
Steering Committee Letter

NORTH Dakota | Water Commission
Be Legendary.™

May 26, 2021

Damon Grabow
Water Resources Engineer
North Dakota State Water Commission
900 East Boulevard Avenue
Bismarck, ND 58505-0850

Subject: North Dakota Statewide PMP Steering Committee Final Report Acceptance

The North Dakota State Water Commission (NDSWC) began the process of developing updated probable maximum precipitation (PMP) data in 2019. This updated analysis was undertaken because the current PMP dataset covering North Dakota was developed in the 1970s and 1980s as part of Hydrometeorological Reports (HMRs) 48 and 51 completed by the National Weather Service and the US Army Corps of Engineers. HMR-51 covered the continental US east of the 105th meridian, while HMR-48 was done specifically for the Red River of the North and the Souris River.

Since the completion of the HMRs, North Dakota has experienced a wet-cycle that was not captured in the data used to develop PMP depths and snowmelt criteria in the previous studies. This period consisted of a number of large spring floods and precipitation events. Many of the historic flooding events in North Dakota occurred due to melting snow or rain on snow events, most recently in 1997, 2009, 2010, and 2011.

This study's purpose is to develop more representative PMP estimates for evaluating flood safety, assessing flood risk, and calibrating event-specific hydrological models. This includes updated summer and spring season PMP depths along with updated snowmelt criteria including snow water equivalent and temperatures times used for snowmelt calculations.

This project is managed by a Steering Committee. The Steering Committee consists of both state and federal agencies with direct knowledge of the sciences and methods involved in a PMP analysis and hydrologic applications. Headed by the NDSWC, the remaining members of the Steering Committee are the National Weather Service (NWS) offices of Bismarck, ND and Grand Forks, ND, the Natural Resources Conservation Services (NRCS) office in Bismarck, ND, the North Dakota State Climatologist at the North Dakota State University (NDSU), and the United States Army Corps of Engineers

(USACE) Omaha District. This board was developed with the intent to guide the NDSWC in selecting a firm to complete the analysis and provide detailed review and comments regarding PMP development and implementation by participating in the program development and selection criteria, maintain the analysis integrity through participation in meetings and discussions, and review the deliverables and final products.

Project tasks and data development were reviewed by the committee. Major tasks are listed below:

- PMP event selection
    - This analysis included a critical review of PMP events in past HMRs and emphasized local meteorological phenomenon. The storm review included extensive analysis of cool season storms.
    - The event selection included a comprehensive list of storm characteristics and descriptive attributes.

- New Storm Analyses
    - Selected storms that were not previously analyzed in adjacent PMP studies were processed through the Storm Precipitation Analysis System (SPAS). This produced standard output that are required for PMP development. These results were reviewed in detail to ensure accuracy.

- Storm Adjustments
    - Selected storms were maximized using updated maximum dew point data. Storms were transposed throughout the state grid system and adjusted for spatial and temporal patterns.

- Storm Transposition
    - Unique transposition limits were applied to each storm used for PMP development. Several iterations and sensitivities were applied to critical storms to test various transposition limits and the resulting effects on PMP depths. Detailed discussions took place to determine the most scientifically accurate and data supported final application of each storm.

- PMP Values
    - PMP values were developed for the state by storm type and season. PMP precipitation values were developed and reviewed for use as PMP products.
    - Numerous iterations of PMP values were developed and reviewed to

determine the most reasonable estimates.

- Snowmelt and Temperature Time Series for spring season PMP
  - Temperature times series were developed over the domain to provide information needed to calculate snowmelt before, during, and after the spring season PMP storm.
  - Snowmelt values were developed statewide. 1% occurrence snow water equivalence values were derived by AWA. These values were made available to be combined with spring season PMP results to derive a total rainfall plus snowmelt runoff.

- Dew Point
  - An updated data set of dew point temperatures was provided as a product of the study.

- Temporal Distribution
  - The analysis included the temporal distribution of PMP events used in the study. The temporal patterns developed within the study include the 90th percentile curve, 50th percentile curve, 10th percentile curve, and the historical storm pattern.

- Spatial Distribution
  - The spatial distribution of the selected PMP events were made available for the PMP products. Historical storms representing typical meteorological patterns have been selected for directional orientation. All of the storms in the study are available.

- GIS Tool
  - A comprehensive GIS tool was provided and reviewed as a product of this study.

Four in-person meetings and several conference calls were held to discuss the progress of the project, analysis results, methods, and processes of the study. All in-person meetings were held in Bismarck, ND.

The Steering Committee agrees that the ND Statewide PMP Analysis consists of a comprehensive review of the most current meteorological precipitation data available

and is conducted according to the latest state of the science technology for determining PMP values. The new PMP data is based upon a more current database of storms from regions appropriate for this study. Updated warm season PMP estimates are an improvement over the previous datasets. The warm season results provide a more robust gridded spatial pattern over the state of North Dakota and rely less heavily on data interpolation over the previous sparsely placed grid points. The spring season results also improve on previous datasets. The spring season PMP data covers the entire state instead of only the Souris and Red River basins and provides statewide 100-year snow water equivalent depths. These data allow the user to compare PMP estimates for the entire runoff season following meteorologically accurate scenarios by storm type and season.

The Steering Committee reviewed the required tasks listed above and accepts the project results for PMP use across the state of North Dakota.

North Dakota State Water Commission

By:

Damon Grabow
Steering Committee Chair
Water Resources Engineer
North Dakota State Water Commission

Date: 05/26/2021

and is conducted according to the latest state of the science technology for determining PMP values. The new PMP data is based upon a more current database of storms from regions appropriate for this study. Updated warm season PMP estimates are an improvement over the previous datasets. The warm season results provide a more robust gridded spatial pattern over the state of North Dakota and rely less heavily on data interpolation over the previous sparsely placed grid points. The spring season results also improve on previous datasets. The spring season PMP data covers the entire state instead of only the Souris and Red River basins and provides statewide 100-year snow water equivalent depths. These data allow the user to compare PMP estimates for the entire runoff season following meteorologically accurate scenarios by storm type and season.

The Steering Committee reviewed the required tasks listed above and accepts the project results for PMP use across the state of North Dakota.

North Dakota State Water Commission

By: ______________________

Damon Grabow
Steering Committee Chair
Water Resources Engineer
North Dakota State Water Commission

Date: 05/26/2021

NORTH Dakota | Water Commission
Be Legendary.™

June 16, 2021

Gregory Gust
Acting Meteorologist-in-Charge
Warning Coordination Meteorologist
National Weather Service
4797 Technology Circle
Grand Forks, ND 58203-0600

Subject: North Dakota Statewide PMP Steering Committee Final Report Acceptance

The North Dakota State Water Commission (NDSWC) began the process of developing updated probable maximum precipitation (PMP) data in 2019. This updated analysis was undertaken because the current PMP dataset covering North Dakota was developed in the 1970s and 1980s as part of Hydrometeorological Reports (HMRs) 48 and 51 completed by the National Weather Service and the US Army Corps of Engineers. HMR-51 covered the continental US east of the 105th meridian, while HMR-48 was done specifically for the Red River of the North and the Souris River.

Since the completion of the HMRs, North Dakota has experienced a wet-cycle that was not captured in the data used to develop PMP depths and snowmelt criteria in the previous studies. This period consisted of a number of large spring floods and precipitation events. Many of the historic flooding events in North Dakota occurred due to melting snow or rain on snow events, most recently in 1997, 2009, 2010, and 2011.

This study's purpose is to develop more representative PMP estimates for evaluating flood safety, assessing flood risk, and calibrating event-specific hydrological models. This includes updated summer and spring season PMP depths along with updated snowmelt criteria including snow water equivalent and temperatures times used for snowmelt calculations.

This project is managed by a Steering Committee. The Steering Committee consists of both state and federal agencies with direct knowledge of the sciences and methods involved in a PMP analysis and hydrologic applications. Headed by the NDSWC, the remaining members of the Steering Committee are the National Weather Service (NWS) offices of Bismarck, ND and Grand Forks, ND, the Natural Resources Conservation

Services (NRCS) office in Bismarck, ND, the North Dakota State Climatologist at the North Dakota State University (NDSU), and the United States Army Corps of Engineers (USACE) Omaha District. This board was developed with the intent to guide the NDSWC in selecting a firm to complete the analysis and provide detailed review and comments regarding PMP development and implementation by participating in the program development and selection criteria, maintain the analysis integrity through participation in meetings and discussions, and review the deliverables and final products.

Project tasks and data development were reviewed by the committee. Major tasks are listed below:

- PMP event selection
  - This analysis included a critical review of PMP events in past HMRs and emphasized local meteorological phenomenon. The storm review included extensive analysis of cool season storms.
  - The event selection included a comprehensive list of storm characteristics and descriptive attributes.

- New Storm Analyses
  - Selected storms that were not previously analyzed in adjacent PMP studies were processed through the Storm Precipitation Analysis System (SPAS). This produced standard output that are required for PMP development. These results were reviewed in detail to ensure accuracy.

- Storm Adjustments
  - Selected storms were maximized using updated maximum dew point data. Storms were transposed throughout the state grid system and adjusted for spatial and temporal patterns.

- Storm Transposition
  - Unique transposition limits were applied to each storm used for PMP development. Several iterations and sensitivities were applied to critical storms to test various transposition limits and the resulting effects on PMP depths. Detailed discussions took place to determine the most scientifically accurate and data supported final application of each storm.

- PMP Values
  - PMP values were developed for the state by storm type and season. PMP precipitation values were developed and reviewed for use as PMP

products.
  - Numerous iterations of PMP values were developed and reviewed to determine the most reasonable estimates.

- Snowmelt and Temperature Time Series for spring season PMP
  - Temperature times series were developed over the domain to provide information needed to calculate snowmelt before, during, and after the spring season PMP storm.
  - Snowmelt values were developed statewide. 1% occurrence snow water equivalence values were derived by AWA. These values were made available to be combined with spring season PMP results to derive a total rainfall plus snowmelt runoff.

- Dew Point
  - An updated data set of dew point temperatures was provided as a product of the study.

- Temporal Distribution
  - The analysis included the temporal distribution of PMP events used in the study. The temporal patterns developed within the study include the 90th percentile curve, 50th percentile curve, 10th percentile curve, and the historical storm pattern.

- Spatial Distribution
  - The spatial distribution of the selected PMP events were made available for the PMP products. Historical storms representing typical meteorological patterns have been selected for directional orientation. All of the storms in the study are available.

- GIS Tool
  - A comprehensive GIS tool was provided and reviewed as a product of this study.

Four in-person meetings and several conference calls were held to discuss the progress of the project, analysis results, methods, and processes of the study. All in-person meetings were held in Bismarck, ND.

The Steering Committee agrees that the ND Statewide PMP Analysis consists of a comprehensive review of the most current meteorological precipitation data available and is conducted according to the latest state of the science technology for determining PMP values. The new PMP data is based upon a more current database of storms from regions appropriate for this study. Updated warm season PMP estimates are an improvement over the previous datasets. The warm season results provide a more robust gridded spatial pattern over the state of North Dakota and rely less heavily on data interpolation over the previous sparsely placed grid points. The spring season results also improve on previous datasets. The spring season PMP data covers the entire state instead of only the Souris and Red River basins and provides statewide 100-year snow water equivalent depths. These data allow the user to compare PMP estimates for the entire runoff season following meteorologically accurate scenarios by storm type and season.

The Steering Committee reviewed the required tasks listed above and accepts the project results for PMP use across the state of North Dakota.

National Weather Service

By:

____

Gregory J. Gust
Acting Meteorologist-in-Charge
Warning Coordination Meteorologist
NOAA/NWS Grand Forks ND

Date: 6/16/2021

NORTH Dakota | Water Commission
Be Legendary.™

May 26, 2021

Jon Petersen, PE
State Hydrologist
Natural Resources Conservation Service
220 East Rosser Avenue
Federal Building, Room 270
Bismarck, ND 58501

Subject: North Dakota Statewide PMP Steering Committee Final Report Acceptance

The North Dakota State Water Commission (NDSWC) began the process of developing updated probable maximum precipitation (PMP) data in 2019. This updated analysis was undertaken because the current PMP dataset covering North Dakota was developed in the 1970s and 1980s as part of Hydrometeorological Reports (HMRs) 48 and 51 completed by the National Weather Service and the US Army Corps of Engineers. HMR-51 covered the continental US east of the 105th meridian, while HMR-48 was done specifically for the Red River of the North and the Souris River.

Since the completion of the HMRs, North Dakota has experienced a wet-cycle that was not captured in the data used to develop PMP depths and snowmelt criteria in the previous studies. This period consisted of a number of large spring floods and precipitation events. Many of the historic flooding events in North Dakota occurred due to melting snow or rain on snow events, most recently in 1997, 2009, 2010, and 2011.

This study's purpose is to develop more representative PMP estimates for evaluating flood safety, assessing flood risk, and calibrating event-specific hydrological models. This includes updated summer and spring season PMP depths along with updated snowmelt criteria including snow water equivalent and temperatures times used for snowmelt calculations.

This project is managed by a Steering Committee. The Steering Committee consists of both state and federal agencies with direct knowledge of the sciences and methods involved in a PMP analysis and hydrologic applications. Headed by the NDSWC, the remaining members of the Steering Committee are the National Weather Service (NWS) offices of Bismarck, ND and Grand Forks, ND, the Natural Resources Conservation Services (NRCS) office in Bismarck, ND, the North Dakota State Climatologist at the

North Dakota State University (NDSU), and the United States Army Corps of Engineers (USACE) Omaha District. This board was developed with the intent to guide the NDSWC in selecting a firm to complete the analysis and provide detailed review and comments regarding PMP development and implementation by participating in the program development and selection criteria, maintain the analysis integrity through participation in meetings and discussions, and review the deliverables and final products.

Project tasks and data development were reviewed by the committee. Major tasks are listed below:

- PMP event selection
    - This analysis included a critical review of PMP events in past HMRs and emphasized local meteorological phenomenon. The storm review included extensive analysis of cool season storms.
    - The event selection included a comprehensive list of storm characteristics and descriptive attributes.

- New Storm Analyses
    - Selected storms that were not previously analyzed in adjacent PMP studies were processed through the Storm Precipitation Analysis System (SPAS). This produced standard output that are required for PMP development. These results were reviewed in detail to ensure accuracy.

- Storm Adjustments
    - Selected storms were maximized using updated maximum dew point data. Storms were transposed throughout the state grid system and adjusted for spatial and temporal patterns.

- Storm Transposition
    - Unique transposition limits were applied to each storm used for PMP development. Several iterations and sensitivities were applied to critical storms to test various transposition limits and the resulting effects on PMP depths. Detailed discussions took place to determine the most scientifically accurate and data supported final application of each storm.

- PMP Values
    - PMP values were developed for the state by storm type and season. PMP precipitation values were developed and reviewed for use as PMP products.

  - Numerous iterations of PMP values were developed and reviewed to determine the most reasonable estimates.

- Snowmelt and Temperature Time Series for spring season PMP
  - Temperature times series were developed over the domain to provide information needed to calculate snowmelt before, during, and after the spring season PMP storm.
  - Snowmelt values were developed statewide. 1% occurrence snow water equivalence values were derived by AWA. These values were made available to be combined with spring season PMP results to derive a total rainfall plus snowmelt runoff.

- Dew Point
  - An updated data set of dew point temperatures was provided as a product of the study.

- Temporal Distribution
  - The analysis included the temporal distribution of PMP events used in the study. The temporal patterns developed within the study include the 90th percentile curve, 50th percentile curve, 10th percentile curve, and the historical storm pattern.

- Spatial Distribution
  - The spatial distribution of the selected PMP events were made available for the PMP products. Historical storms representing typical meteorological patterns have been selected for directional orientation. All of the storms in the study are available.

- GIS Tool
  - A comprehensive GIS tool was provided and reviewed as a product of this study.

Four in-person meetings and several conference calls were held to discuss the progress of the project, analysis results, methods, and processes of the study. All in-person meetings were held in Bismarck, ND.

The Steering Committee agrees that the ND Statewide PMP Analysis consists of a comprehensive review of the most current meteorological precipitation data available and is conducted according to the latest state of the science technology for determining PMP values. The new PMP data is based upon a more current database of storms from regions appropriate for this study. Updated warm season PMP estimates are an improvement over the previous datasets. The warm season results provide a more robust gridded spatial pattern over the state of North Dakota and rely less heavily on data interpolation over the previous sparsely placed grid points. The spring season results also improve on previous datasets. The spring season PMP data covers the entire state instead of only the Souris and Red River basins and provides statewide 100-year snow water equivalent depths. These data allow the user to compare PMP estimates for the entire runoff season following meteorologically accurate scenarios by storm type and season.

The Steering Committee reviewed the required tasks listed above and accepts the project results for PMP use across the state of North Dakota.

NATURAL RESOURCES CONSERVATION SERVICE

By: Jonathan Petersen

Jon Petersen, P.E.
Hydrologist
NRCS, Bismarck State Office

Date: 6-16-2021

NORTH Dakota | Water Commission
Be Legendary.

May 26, 2021

John Paul Martin
Warning Coordination Meteorologist
National Weather Service
2301 University Drive, Building 27
Bismarck, ND 58504

Subject: North Dakota Statewide PMP Steering Committee Final Report Acceptance

The North Dakota State Water Commission (NDSWC) began the process of developing updated probable maximum precipitation (PMP) data in 2019. This updated analysis was undertaken because the current PMP dataset covering North Dakota was developed in the 1970s and 1980s as part of Hydrometeorological Reports (HMRs) 48 and 51 completed by the National Weather Service and the US Army Corps of Engineers. HMR-51 covered the continental US east of the 105th meridian, while HMR-48 was done specifically for the Red River of the North and the Souris River.

Since the completion of the HMRs, North Dakota has experienced a wet-cycle that was not captured in the data used to develop PMP depths and snowmelt criteria in the previous studies. This period consisted of a number of large spring floods and precipitation events. Many of the historic flooding events in North Dakota occurred due to melting snow or rain on snow events, most recently in 1997, 2009, 2010, and 2011.

This study's purpose is to develop more representative PMP estimates for evaluating flood safety, assessing flood risk, and calibrating event-specific hydrological models. This includes updated summer and spring season PMP depths along with updated snowmelt criteria including snow water equivalent and temperatures times used for snowmelt calculations.

This project is managed by a Steering Committee. The Steering Committee consists of both state and federal agencies with direct knowledge of the sciences and methods involved in a PMP analysis and hydrologic applications. Headed by the NDSWC, the remaining members of the Steering Committee are the National Weather Service (NWS) offices of Bismarck, ND and Grand Forks, ND, the Natural Resources Conservation Services (NRCS) office in Bismarck, ND, the North Dakota State Climatologist at the North Dakota State University (NDSU), and the United States Army Corps of Engineers (USACE) Omaha District. This board was developed with the intent to guide the NDSWC in selecting a firm to complete the analysis and provide detailed review and comments regarding PMP development and implementation by participating in the program development and selection criteria, maintain the analysis integrity through participation in meetings and discussions, and review the deliverables and final products.

Project tasks and data development were reviewed by the committee. Major tasks are listed below:

- PMP event selection
  - This analysis included a critical review of PMP events in past HMRs and emphasized local meteorological phenomenon. The storm review included extensive analysis of cool season storms.
  - The event selection included a comprehensive list of storm characteristics and descriptive attributes.

- New Storm Analyses
  - Selected storms that were not previously analyzed in adjacent PMP studies were processed through the Storm Precipitation Analysis System (SPAS). This produced standard output that are required for PMP development. These results were reviewed in detail to ensure accuracy.

- Storm Adjustments
  - Selected storms were maximized using updated maximum dew point data. Storms were transposed throughout the state grid system and adjusted for spatial and temporal patterns.

- Storm Transposition
  - Unique transposition limits were applied to each storm used for PMP development. Several iterations and sensitivities were applied to critical storms to test various transposition limits and the resulting effects on PMP depths. Detailed discussions took place to determine the most scientifically accurate and data supported final application of each storm.

- PMP Values
  - PMP values were developed for the state by storm type and season. PMP precipitation values were developed and reviewed for use as PMP products.
  - Numerous iterations of PMP values were developed and reviewed to determine the most reasonable estimates.

- Snowmelt and Temperature Time Series for spring season PMP
  - Temperature times series were developed over the domain to provide information needed to calculate snowmelt before, during, and after the spring season PMP storm.
  - Snowmelt values were developed statewide. 1% occurrence snow water equivalence values were derived by AWA. These values were made available to be combined with spring season PMP results to derive a total

rainfall plus snowmelt runoff.

- Dew Point
  - An updated data set of dew point temperatures was provided as a product of the study.

- Temporal Distribution
  - The analysis included the temporal distribution of PMP events used in the study. The temporal patterns developed within the study include the 90th percentile curve, 50th percentile curve, 10th percentile curve, and the historical storm pattern.

- Spatial Distribution
  - The spatial distribution of the selected PMP events were made available for the PMP products. Historical storms representing typical meteorological patterns have been selected for directional orientation. All of the storms in the study are available.

- GIS Tool
  - A comprehensive GIS tool was provided and reviewed as a product of this study.

Four in-person meetings and several conference calls were held to discuss the progress of the project, analysis results, methods, and processes of the study. All in-person meetings were held in Bismarck, ND.

The Steering Committee agrees that the ND Statewide PMP Analysis consists of a comprehensive review of the most current meteorological precipitation data available and is conducted according to the latest state of the science technology for determining PMP values. The new PMP data is based upon a more current database of storms from regions appropriate for this study. Updated warm season PMP estimates are an improvement over the previous datasets. The warm season results provide a more robust gridded spatial pattern over the state of North Dakota and rely less heavily on data interpolation over the previous sparsely placed grid points. The spring season results also improve on previous datasets. The spring season PMP data covers the entire state instead of only the Souris and Red River basins and provides statewide 100-year snow water equivalent depths. These data allow the user to compare PMP estimates for the entire runoff season following meteorologically accurate scenarios by storm type and season.

The Steering Committee reviewed the required tasks listed above and accepts the project results for PMP use across the state of North Dakota.

National Weather Service

By:

John Paul Martin

JOHN PAUL MARTIN
WCM – Warning Coordination Meteorologist
NOAA/NWS Bismarck ND

Date: JUNE 2, 2021

NORTH Dakota | Water Commission

Be Legendary.™

June 4, 2021

Matthew Masek
US Army Corps of Engineers, Omaha District
1616 Capitol Ave., Ste. 9000
Omaha, NE 68102

Subject: North Dakota Statewide PMP Steering Committee Final Report Acceptance

The North Dakota State Water Commission (NDSWC) began the process of developing updated probable maximum precipitation (PMP) data in 2019. This updated analysis was undertaken because the current PMP dataset covering North Dakota was developed in the 1970s and 1980s as part of Hydrometeorological Reports (HMRs) 48 and 51 completed by the National Weather Service and the US Army Corps of Engineers. HMR-51 covered the continental US east of the 105th meridian, while HMR-48 was done specifically for the Red River of the North and the Souris River.

Since the completion of the HMRs, North Dakota has experienced a wet-cycle that was not captured in the data used to develop PMP depths and snowmelt criteria in the previous studies. This period consisted of a number of large spring floods and precipitation events. Many of the historic flooding events in North Dakota occurred due to melting snow or rain on snow events, most recently in 1997, 2009, 2010, and 2011.

This study's purpose is to develop more representative PMP estimates for evaluating flood safety, assessing flood risk, and calibrating event-specific hydrological models. This includes updated summer and spring season PMP depths along with updated snowmelt criteria including snow water equivalent and temperatures times used for snowmelt calculations.

This project is managed by a Steering Committee. The Steering Committee consists of both state and federal agencies with direct knowledge of the sciences and methods involved in a PMP analysis and hydrologic applications. Headed by the NDSWC, the remaining members of the Steering Committee are the National Weather Service (NWS) offices of Bismarck, ND and Grand Forks, ND, the Natural Resources Conservation Services (NRCS) office in Bismarck, ND, the North Dakota State Climatologist at the North Dakota State University (NDSU), and the United States Army Corps of Engineers (USACE) Omaha District. This board was developed with the intent to guide the NDSWC in selecting a firm to complete the analysis and provide detailed review and comments regarding PMP development and implementation by participating in the program development and selection criteria, maintain the analysis integrity through participation in meetings and discussions, and review the deliverables and final products.

Project tasks and data development were reviewed by the committee. Major tasks are listed below:

- PMP event selection
  - This analysis included a critical review of PMP events in past HMRs and emphasized local meteorological phenomenon. The storm review included extensive analysis of cool season storms.
  - The event selection included a comprehensive list of storm characteristics and descriptive attributes.

- New Storm Analyses
  - Selected storms that were not previously analyzed in adjacent PMP studies were processed through the Storm Precipitation Analysis System (SPAS). This produced standard output that are required for PMP development. These results were reviewed in detail to ensure accuracy.

- Storm Adjustments
  - Selected storms were maximized using updated maximum dew point data. Storms were transposed throughout the state grid system and adjusted for spatial and temporal patterns.

- Storm Transposition
  - Unique transposition limits were applied to each storm used for PMP development. Several iterations and sensitivities were applied to critical storms to test various transposition limits and the resulting effects on PMP depths. Detailed discussions took place to determine the most scientifically accurate and data supported final application of each storm.

- PMP Values
  - PMP values were developed for the state by storm type and season. PMP precipitation values were developed and reviewed for use as PMP products.
  - Numerous iterations of PMP values were developed and reviewed to determine the most reasonable estimates.

- Snowmelt and Temperature Time Series for spring season PMP
    - Temperature times series were developed over the domain to provide information needed to calculate snowmelt before, during, and after the spring season PMP storm.
    - Snowmelt values were developed statewide. 1% occurrence snow water equivalence values were derived by AWA. These values were made available to be combined with spring season PMP results to derive a total rainfall plus snowmelt runoff.

- Dew Point
    - An updated data set of dew point temperatures was provided as a product of the study.

- Temporal Distribution
    - The analysis included the temporal distribution of PMP events used in the study. The temporal patterns developed within the study include the 90th percentile curve, 50th percentile curve, 10th percentile curve, and the historical storm pattern.

- Spatial Distribution
    - The spatial distribution of the selected PMP events were made available for the PMP products. Historical storms representing typical meteorological patterns have been selected for directional orientation. All of the storms in the study are available.

- GIS Tool
    - A comprehensive GIS tool was provided and reviewed as a product of this study.

Four in-person meetings and several conference calls were held to discuss the progress of the project, analysis results, methods, and processes of the study. All in-person meetings were held in Bismarck, ND.

The Steering Committee agrees that the ND Statewide PMP Analysis consists of a comprehensive review of the most current meteorological precipitation data available and is conducted according to the latest state of the science technology for determining

PMP values. The new PMP data is based upon a more current database of storms from regions appropriate for this study. Updated warm season PMP estimates are an improvement over the previous datasets. The warm season results provide a more robust gridded spatial pattern over the state of North Dakota and rely less heavily on data interpolation over the previous sparsely placed grid points. The spring season results also improve on previous datasets. The spring season PMP data covers the entire state instead of only the Souris and Red River basins and provides statewide 100-year snow water equivalent depths. These data allow the user to compare PMP estimates for the entire runoff season following meteorologically accurate scenarios by storm type and season.

The Steering Committee reviewed the required tasks listed above and accepts the project results for PMP use across the state of North Dakota.

United States Army Corps of Engineers

By:

MASEK.MATTHEW.J AMES.1365876963 Digitally signed by MASEK.MATTHEW.JAMES.1365876963 Date: 2021.06.04 17:09:10 -05'00'

Matthew Masek
Hydrologic Hazards Team - Meteorologist
U.S. Army Corps of Engineers, Omaha District

Date: 06/04/2021

NORTH Dakota | Water Commission
Be Legendary.™

May 26, 2021

Adnan Akyuz, Ph.D.
North Dakota State Climatologist
NDSU Department 7521, PO Box 6050
Fargo, ND 58108-6050

Subject: North Dakota Statewide PMP Steering Committee Final Report Acceptance

The North Dakota State Water Commission (NDSWC) began the process of developing updated probable maximum precipitation (PMP) data in 2019. This updated analysis was undertaken because the current PMP dataset covering North Dakota was developed in the 1970s and 1980s as part of Hydrometeorological Reports (HMRs) 48 and 51 completed by the National Weather Service and the US Army Corps of Engineers. HMR-51 covered the continental US east of the 105th meridian, while HMR-48 was done specifically for the Red River of the North and the Souris River.

Since the completion of the HMRs, North Dakota has experienced a wet-cycle that was not captured in the data used to develop PMP depths and snowmelt criteria in the previous studies. This period consisted of a number of large spring floods and precipitation events. Many of the historic flooding events in North Dakota occurred due to melting snow or rain on snow events, most recently in 1997, 2009, 2010, and 2011.

This study's purpose is to develop more representative PMP estimates for evaluating flood safety, assessing flood risk, and calibrating event-specific hydrological models. This includes updated summer and spring season PMP depths along with updated snowmelt criteria including snow water equivalent and temperatures times used for snowmelt calculations.

This project is managed by a Steering Committee. The Steering Committee consists of both state and federal agencies with direct knowledge of the sciences and methods involved in a PMP analysis and hydrologic applications. Headed by the NDSWC, the remaining members of the Steering Committee are the National Weather Service (NWS) offices of Bismarck, ND and Grand Forks, ND, the Natural Resources Conservation Services (NRCS) office in Bismarck, ND, the North Dakota State Climatologist at the North Dakota State University (NDSU), and the United States Army Corps of Engineers

(USACE) Omaha District. This board was developed with the intent to guide the NDSWC in selecting a firm to complete the analysis and provide detailed review and comments regarding PMP development and implementation by participating in the program development and selection criteria, maintain the analysis integrity through participation in meetings and discussions, and review the deliverables and final products.

Project tasks and data development were reviewed by the committee. Major tasks are listed below:

- PMP event selection
  - This analysis included a critical review of PMP events in past HMRs and emphasized local meteorological phenomenon. The storm review included extensive analysis of cool season storms.
  - The event selection included a comprehensive list of storm characteristics and descriptive attributes.

- New Storm Analyses
  - Selected storms that were not previously analyzed in adjacent PMP studies were processed through the Storm Precipitation Analysis System (SPAS). This produced standard output that are required for PMP development. These results were reviewed in detail to ensure accuracy.

- Storm Adjustments
  - Selected storms were maximized using updated maximum dew point data. Storms were transposed throughout the state grid system and adjusted for spatial and temporal patterns.

- Storm Transposition
  - Unique transposition limits were applied to each storm used for PMP development. Several iterations and sensitivities were applied to critical storms to test various transposition limits and the resulting effects on PMP depths. Detailed discussions took place to determine the most scientifically accurate and data supported final application of each storm.

- PMP Values
  - PMP values were developed for the state by storm type and season. PMP precipitation values were developed and reviewed for use as PMP products.
  - Numerous iterations of PMP values were developed and reviewed to

and is conducted according to the latest state of the science technology for determining PMP values. The new PMP data is based upon a more current database of storms from regions appropriate for this study. Updated warm season PMP estimates are an improvement over the previous datasets. The warm season results provide a more robust gridded spatial pattern over the state of North Dakota and rely less heavily on data interpolation over the previous sparsely placed grid points. The spring season results also improve on previous datasets. The spring season PMP data covers the entire state instead of only the Souris and Red River basins and provides statewide 100-year snow water equivalent depths. These data allow the user to compare PMP estimates for the entire runoff season following meteorologically accurate scenarios by storm type and season.

The Steering Committee reviewed the required tasks listed above and accepts the project results for PMP use across the state of North Dakota.

NORTH DAKOTA STATE UNIVERSITY

By: ______________________________

Adnan Akyuz
North Dakota State Climatologist
NDSU

Date: 5/27/2021