Workpapers of Chapter 2 (Wei Bin Guo)

Application: A.22-09-015

SoCalGas and SDG&E

2024 Cost Allocation Proceeding

(WEATHER DESIGN)

Table of Contents

Weather for SoCalGas: Heating Degree Days – Average and Cold Year Designs; and Winter Peak Day Design Temperatures

September 2022

I. Overview

Southern California Gas Company's service area extends from Fresno County to the Mexican border. To quantify the overall temperature experienced within this region, SoCalGas aggregates daily temperature recordings from fifteen U.S. Weather Bureau weather stations first into six temperature zones and then into one system average heating degree-day ("HDD") figure. The table below lists weather station locations by temperature zones.

Table 1

Weather Stations by Temperature Zones and Weights

Temperature Zone	Weight	Station (After 10/31/2002)	Station (Before 11/1/2002)
1. High mountain	0.0057	Big Bear Lake	Lake Arrowhead
2. Low desert	0.0386	Palm Springs El Centro	Palm Springs Brawley
3. Coastal	0.1821	Los Angeles Airport Newport Beach Santa Barbara Airport	Los Angeles Airport Newport Beach Harbor Santa Barbara Airport
4. High desert	0.0722	Bakersfield Lancaster Airport Fresno	Bakersfield Airport Palmdale Visalia
5. Interior valleys	0.3819	Burbank Pasadena Ontario Rialto	Burbank Pasadena Pomona Cal Poly Redlands
6. Basin	0.3195	Los Angeles Civic Center Santa Ana	Los Angeles Civic Center/ Downtown-USC Santa Ana

SoCalGas uses 65° Fahrenheit to calculate the number of HDDs. One heating degree day is accumulated for each degree that the daily average is below 65° Fahrenheit. To arrive at the HDD figure for each temperature zone, SoCalGas uses the simple average of the weather station HDDs in that temperature zone. To arrive at the system average HDDs figure for its entire service area, SoCalGas weights the HDD figure for each zone using the proportion of gas customers within each temperature zone based on December 2021 customer counts. These weights have been used in calculating the data shown from January 2002 to December 2021.

Daily weather temperatures are from the National Climatic Data Center or from preliminary data that SoCalGas captures each day and posts on its internal Company server directory at \\ap-ewerep-p01\weather\b_detail\ for various individual weather stations as well as for its system average values of HDD. Annual HDDs for the entire service area from 2002 to 2021 are listed in Table 2, below.

Table 2

Calendar Month Heating Degree-Days (Jan. 2002 through Dec. 2021)

	Month												Total
Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	"Cal-Year"
2002	334	202	226	148	78	10	2	4	8	77	92	315	**1496**
2003	141	232	166	180	74	17	1	1	3	16	200	306	**1337**
2004	292	301	86	85	17	8	3	2	4	73	227	292	**1390**
2005	287	208	176	115	35	11	4	1	9	44	99	235	**1224**
2006	272	200	338	162	28	3	0	1	5	36	104	278	**1427**
2007	347	214	125	117	50	16	1	1	12	37	126	353	**1399**
2008	347	262	148	123	76	8	1	0	2	23	75	334	**1399**
2009	196	259	194	134	18	16	3	4	1	43	117	320	**1305**
2010	254	220	173	164	71	14	8	9	14	42	203	268	**1440**
2011	250	307	211	105	80	27	3	3	6	39	207	349	**1587**
2012	224	236	222	118	38	11	6	1	1	16	110	300	**1283**
2013	329	263	125	65	17	4	1	2	2	44	103	257	**1212**
2014	142	148	90	76	19	4	0	1	1	5	66	223	**775**
2015	180	94	64	67	69	5	1	0	1	4	162	316	**963**
2016	281	111	113	54	45	8	1	1	3	14	110	268	**1009**
2017	319	208	99	44	50	6	1	0	4	12	50	174	**967**
2018	155	210	180	71	57	6	0	0	1	10	79	247	**1016**
2019	262	350	165	53	76	9	2	1	3	23	125	264	**1333**
2020	241	174	204	107	11	3	3	2	1	10	149	236	**1141**
2021	258	180	231	74	37	8	0	1	9	41	74	336	**1249**
20-Yr-Avg (Jan2002-Dec2021)													
Avg.	255.6	219.0	166.8	103.1	47.3	9.7	2.1	1.8	4.5	30.5	123.9	283.6	**1247.6**
St.Dev.	65.8	62.6	64.4	40.5	24.1	5.9	2.1	2.1	3.9	20.8	51.4	46.9	**210.1**
Min.	141.0	94.0	64.0	44.0	11.0	3.0	0.0	0.0	1.0	4.0	50.0	174.0	**775.0**
Max.	347.0	350.0	338.0	180.0	80.0	27.0	8.0	9.0	14.0	77.0	227.0	353.0	**1587.0**

II. Calculations to Define Our Average-Temperature Year

The simple average of the 20-year period (January 2002 through December 2021) was used to represent the Average Year total and the individual monthly values for HDD. In this CGR, the standard deviation has been calculated using an approach that compensates for the annual HDD values for the years 2014-2018 in SoCalGas' service territory being dramatically lower than in any preceding year going back to 1950. A regression with a time trend and a dummy variable for the years 2014-2018 has been used to estimate a shift in the level of annual HDD that occurred beginning in 2014. A dummy variable takes the value one for some observations to indicate the presence of an effect or membership in a group and zero for the remaining observations. Estimating the effect of the dummy variable gives an estimate of that effect or the impact of membership in that group. A dummy variable is used here to estimate the average effect on annual HDD of a given year having membership in the group of years 2014-2018. The dataset is SoCalGas system-wide annual HDD for the years 2002-2021. The regression equation is:

$$HDD_t = \alpha + \beta * t + \beta_{2014-2018} * D_{2014-2018} + \varepsilon$$

where $D_{2014-2018}$ is a dummy variable for the years 2014-2018 and $\beta_{2014-2018}$is the corresponding dummy coefficient. This regression equation estimates average HDD over the period 2002-2021 controlling for time trends in HDD and the warm weather regime of years 2014-2018. It's important to note that p-value for the estimate of $\beta_{2014-2018}$ is less than 0.003%, indicating an extremely low probability that membership in the group of years 2014-2018 had no effect on annual HDDs. Please see table 3 below for the full regression output.

Table 3

Dummy Regression for Calculation of Heating Degree-Day Standard Deviation

Regression Statistics	
Multiple R	0.875160819
R Square	0.765906459
Adjusted R Squar	0.738366043
Standard Error	107.4790962
Observations	20

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	642514.9459	321257.473	27.8102714	4.36325E-06
Residual	17	196379.8541	11551.75612		
Total	19	838894.8			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	17936.32201	9383.914185	1.911390243	0.072963703
Regime Dummy	-352.6163522	62.17005618	-5.67180366	0.000027572
YEAR	-8.252830189	4.668593607	-1.767733687	0.095048045

The dummy variable's estimated effect, $\beta_{2014-2018}$, is subtracted from the actual annual HDD data for years 2014-2018 to adjust the data to remove the level shift. The standard deviation has been calculated using this adjusted dataset. This standard deviation has been used to design the two Cold Years based on a “1-in-10” and “1-in-35” chance, ***c***, that the respective annual “Cold Year” $\boldsymbol{hdd_c}$ value would be exceeded.

A probability model for the annual HDD is based on a t-Distribution with N-1 degrees of freedom, where, N is the number of years of HDD data we use, μ is the average of the last 20 years of HDD, and S_{20} is the average of the standard deviations of the 20 most recent 20 year periods:

$U = (HDD_y - \mu)/S_{20}$, has a t-Distribution with N-1 degrees of freedom.

III. Calculating the Cold-Temperature Year Weather Designs

Cold Year HDD Weather Designs

For SoCalGas, cold-temperature-year HDD weather designs are developed with a 1-in-35 annual chance of occurrence. In terms of probabilities this can be expressed as the following for a “1-in-35” cold-year HDD value in equation 1 and a “1-in-10” cold-year HDD value in equation 2, with Annual HDD as the random variable:

(1) Prob { Annual HDD > “1-in-35” Cold-Yr HDD } = 1/35 = 0.0286

(2) Prob { Annual HDD > “1-in-10” Cold-Yr HDD } = 1/10 = 0.1000

An area of 0.0286 under one tail of the T-Distribution translates to 2.025 standard deviations *above* an average-year based on a t-statistic with 19 degrees of freedom. Using the standard deviation calculated as described earlier, which is 112.8 HDD, these equations yield values of about 1,476 HDD for a “1-in-35” cold year and 1,398 HDDs for a “1-in-10” cold year. (An area of 0.1000 under one tail of the T-Distribution translates to 1.328 standard deviations *above* an average-year based on a t-statistic with 19 degrees of freedom.) For example, the “1-in-35” cold-year HDD is calculated as follows:

(3) Cold-year HDD = 1,476 which equals approximately
1,248 average-year HDDs + 2.025 * 112.8

Table 4 shows monthly HDD figures for “1-in-35” cold year, “1-in-10” cold year and, average year temperature designs. The monthly average-temperature-year HDDs are calculated from weighted monthly HDDs from 2002 to 2021, as shown as the bottom of Table 2, above. For example, the average-year December value of 283.6 HDD equals the simple average of the twenty

December HDD figures from 2002 to 2021. SoCalGas calculates the cold--temperature-year monthly HDD values using the same distribution of average-year HDDs. For example, 22.73 percent (283.6 / 1247.6) of average-temperature-year HDDs occurred in December, so the estimated number of HDDs during December for a 1-in-35 cold-year is equal to 1,476 HDDs multiplied by 22.73 percent, or 335.5 HDDs.

Table 4

Calendar Month Heating Degree-Day Designs

	Cold		Average	Hot	
	1-in-35 Design	**1-in-10 Design**		**1-in-10 Design**	**1-in-35 Design**
January	302.3	286.4	255.6	224.9	208.9
February	259.0	245.3	219.0	192.7	179.0
March	197.3	186.9	166.9	146.8	136.4
April	122.0	115.5	103.1	90.7	84.3
May	56.0	53.0	47.3	41.6	38.7
June	11.5	10.9	9.7	8.5	7.9
July	2.4	2.3	2.1	1.8	1.7
August	2.1	2.0	1.8	1.5	1.4
September	5.3	5.0	4.5	4.0	3.7
October	36.0	34.1	30.5	26.8	24.9
November	146.6	138.8	123.9	109.0	101.3
December	335.5	317.7	283.6	249.5	231.8
	1476	1398	1248	1098	1020

IV. Adjusting Forecasted HDDs for a Climate-Change Trend

SoCalGas incorporates a climate-change warming trend that reduces HDDs by 6 HDDs per year over the forecast period. The annual reduction is based on the latest twenty-year trend in 20-year-averaged HDDs. That is, they are based on the observed trend in changes starting with average HDDs for years 1983-2002, then 1984-2003, 1985-2004...and ending with the average HDDs for years 2002-2021.

Table 5 below shows system HDDs, rolling 20-year averaged HDDs, and the annual changes in those rolling 20-year averages. The actual average annual change is -7.0 HDDs for the most recent twenty of the 20-year averages (with ending years from 2002 through 2021). A simple “ordinary least squares” regression-fitted time trend (using Microsoft Excel’s “LINEST” function) was applied to those same annual changes, resulting in a fitted estimation of -6.2 HDDs per year. Based on the fitted trend, it was decided to decrease average-year and cold-year forecasted HDD’s by an even 6 HDDs per year, starting with the first forecast year of 2022.

Table 5

Average Annual Changes in 20-Year Averaged Heating-Degree Days

Average Annual Changes in 20-Year-Averaged HDDs		
	Regression Fitted trend	Actual
20 years (2002-2021)	-6.2	-7.0

Year	SoCalGas System HDDs	20-year averaged HDDs	Annual change in 20-year averaged HDDs
1982	1650		
1983	1384		
1984	1332		
1985	1584		
1986	1090		
1987	1497		
1988	1365		
1989	1359		
1990	1441		
1991	1405		
1992	1252		
1993	1208		
1994	1462		
1995	1240		
1996	1183		
1997	1152		
1998	1565		
1999	1535		
2000	1369		
2001	1688	1388.1	
2002	1496	1380.4	-7.7
2003	1337	1378.0	-2.3
2004	1390	1380.9	2.9
2005	1224	1362.9	-18.0
2006	1427	1379.8	16.8
2007	1399	1374.9	-4.9
2008	1399	1376.6	1.7
2009	1305	1373.9	-2.7
2010	1440	1373.8	0.0
2011	1587	1382.9	9.1
2012	1283	1384.5	1.5
2013	1212	1384.7	0.2
2014	775	1350.3	-34.4
2015	963	1336.5	-13.8
2016	1009	1327.8	-8.7
2017	967	1318.5	-9.3
2018	1016	1291.1	-27.5
2019	1333	1281.0	-10.1
2020	1141	1269.6	-11.4
2021	1249	1247.6	-22.0

V. Calculating the Peak-Day Design Temperature

SoCalGas' 1-in-35 Peak-Day design temperature of 40.5 degrees Fahrenheit, denoted "Deg-F," is determined from a statistical analysis of observed annual minimum daily system average temperatures constructed from daily temperature recordings from the fifteen U.S. Weather Bureau weather stations discussed above. Since we have a time series of daily data by year, the following notation will be used for the remainder of this discussion:

(1) $AVG_{y,d}$ = sy stem avg value of temperature for calendar year "y" and day "d".

The calendar year, y, can range from 1950 through 2021, while the day, d, can range from 1 to 365, for non-leap years, or from 1 to 366 for leap years. The "upper" value for the day, d, thus depends on the calendar year, y, and will be denoted by n(y)=365, or 366, respectively, when y is a non-leap year or a leap year.

For each calendar year, we calculate the following statistic from our series of daily system average temperatures defined in equation (1) above:

$$(2) \qquad MinAVG_y = \min_{d=1}^{n(y)}\{ AVG_{y,d} \}, \text{ for } y=1950, 1951, \ldots, 2021.$$

(The notation used in equation 2 means "For a particular year, y, list all the daily values of system average temperature for that year, then pick the smallest one.")

The resulting minimum annual temperatures are shown in Tables 6.1 and 6.2, below. Most of the minimum temperatures occur in the months of December, January, or February; for a few calendar years the minimums occurred in March or November.

The statistical methods we use to analyze this data employ software developed to fit three generic probability models: the Generalized Extreme Value (GEV) model, the Double-Exponential or GUMBEL (EV1) model and a 2-Parameter Students' T-Distribution (T-Dist) model. [The GEV and EV1 models have the same mathematical specification as those implemented in a DOS-based executable-only computer code that was developed by Richard L. Lehman and described in a paper published in the Proceedings of the Eighth Conference on Applied Climatology, January 17-22, 1993, Anaheim, California, pp. 270-273, by the American Meteorological Society, Boston, MA., with the title "Two Software Products for Extreme Value Analysis: System Overviews of ANYEX and DDEX." At the time he wrote the paper, Dr. Lehman was with the Climate Analysis Center, National Weather Service/NOAA in Washington, D.C., zip code 20233.] The Statistical Analysis Software (SAS) procedure for nonlinear statistical model estimation (PROC MODEL) was used to do the calculations. Further, the calculation procedures were implemented to fit the probability models to observed *maxima* of data, like heating degrees. By recognizing that:

$$-\ MinAVG_y = -\min_{d=1}^{n(y)}\{AVG_{y,d}\} = \max_{d=1}^{n(y)}\{-AVG_{y,d}\},\ \text{for } y=1950, \ldots, 2021$$

this same software, when applied to the *negative* of the minimum temperature data, yields appropriate probability model estimation results.

The calculations done to fit any one of the three probability models chooses the parameter values that provide the "best fit" of the parametric probability model's calculated cumulative distribution function (CDF) to the empirical cumulative distribution function (ECDF). Note that the ECDF is constructed based on the variable "$-MinAVG_y$" (which is a *maximum* over a set of *negative* temperatures) with values of the variable $MinAVG_y$ that are the same as shown in Tables 6.1 and 6.2, below.

In Tables 7.1 and 7.2, the data for $-MinAVG_y$ are shown after they have been sorted from "lowest" to "highest" value. The ascending *ordinal* value is shown in the column labeled "RANK" and the empirical cumulative distribution function is calculated and shown in the next column. The formula used to calculate this function is:

$$ECDF = (RANK - \alpha)/[MaxRANK + (1 - 2\alpha)],$$

where the parameter "α" (shown as *alpha* in Table 7.1 and Table 7.2) is a "small" positive value (usually less than ½) that is used to bound the ECDF away from 0 and 1.

Of the three probability models considered (GEV, EV1, and T_Dist) the results obtained for the T_Dist model were selected since the fit to the ECDF was better than that of either the GEV model or the EV1 model. (Although convergence to stable parameter estimates is occasionally a problem with fitting a GEV model to the ECDF, the T_Dist model had no problems with convergence of the iterative procedure to estimate parameters.)

The T_Dist model used here is a three-parameter probability model where the varable $z = (-MinAVG_y - \gamma) / \theta$, for each year, y, is presumed to follow a T_Dist with location parameter, γ, and scale parameter, θ, and a third parameter, ν, that represents the number of degrees of freedom. For a given number of years of data, N, then ν=N-2.

The following mathematical expression specifies the T_Dist model we fit to the data for "$-MinAVG_y$ " shown in Table 7.1 and Table 7.2, below.

(3) $ECDF(-MinAVG_y) = Prob\{-T < -MinAVG_y\} = T\_Dist\{z; \gamma, \theta, \nu=N-2\}$,

where "T_Dist{ . }" is the cumulative probability distribution function for Student's T-Distribution[1], and

[1] A common mathematical expression for Student's T-Distribution is provided at http://en.wikipedia.org/wiki/Student%27s_t-distribution; with a probability density function

$$f(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\,\Gamma(\frac{\nu}{2})}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

(4) $z = (-MinAVG_y - \gamma) / \theta$, for each year, y, and

the parameters "γ" and "θ" are estimated for this model for given degrees of freedom v=N-2. The estimated values for γ and θ are shown in Table 7.2 along with the fitted values of the model CDF (the column: "Fitted" Model CDF).

Now, to calculate a *peak-day design temperature*, $TPDD_\delta$, with a specified likelihood, δ, that a value less than $TPDD_\delta$ would be observed, we use the equation below:

(5) $\delta = \text{Prob}\ \{ T \leq TPDD_\delta \}$, which is equivalent to

(6) $\delta = \text{Prob}\ \{ [(-T - \gamma) / \theta] \geq [(-TPDD_\delta - \gamma) / \theta] \}, = \text{Prob}\ \{ [(-T - \gamma) / \theta] \geq [z_\delta] \}$,

where $z_\delta = [(-TPDD_\delta - \gamma) / \theta]$. In terms of our probability model,

(7) $\delta = 1 - \text{T\_Dist}\{ z_\delta; \gamma, \theta, v=N-2\}$,

which yields the following equation for z_δ,

(7') $z_\delta = \{ \text{TINV\_Dist}\{ (1-\delta); \gamma, \theta, v=N-2\}$, where "TINV_Dist{ . }" is the inverse function of the T_Dist{ . } function[2]. The implied equation for $TPDD_\delta$ is:

(8) $TPDD_\delta = - [\gamma + (z_\delta)(\theta)]$.

To calculate the minimum daily (system average) temperature to define our extreme weather event, we specify that this COLDEST-Day be one where the temperature would be lower with a "1-in-35" likelihood. This criterion translates into two equations to be solved based on equations (7) and (8) above:

(9) solve for "z_δ" from equation (7') above with (1-δ) = (1 - 1/35) = 1 - 0.0286,

(10) solve for "$TPDD_\delta$" from $TPDD_\delta = - [\gamma + (z_\delta)(\theta)]$.

The value of $z_\delta = 1.935$ and $TPDD_\delta = - [\gamma + (z_\delta)(\theta)] = 40.5$ degrees Fahrenheit, with values for "v=N-2"; along with "γ" and "θ" in Tables 7.1 & 7.2, below.

SoCalGas' 1-in-10 peak-day design temperature of 42.2 degrees Fahrenheit, is calculated in a methodologically similar way as for the 40.5 degree peak day temperature. The criteria specified in equation (9) above for a "1-in-35" likelihood would be replaced by a "1-in-10" likelihood.

(9') solve for "z_δ" from equation (7') above with (1-δ) = (1 - 1/10) = 1 - 0.1000,

which yields a "z_δ" value of $z_\delta = 1.294$ and, $TPDD_\delta = - [\gamma + (z_\delta)(\theta)] = 42.2$ with values for "v=N-2"; along with "γ" and "θ" in Tables 7.1 and 7.2, below.

A plot of the cumulative distribution function for $MinAVG_y$ based on "v=N-2", the fitted model parameters, "γ" and "θ" with values in Tables 7.1 and 7.2, below, is shown in Figure 1.

such that $\text{T\_Dist}\{z; \gamma, \theta, \nu=N-2\}=\int f(t)\,dt$, from $t=-\infty$ to $t=z$. Also, the notation Γ(.) is known in mathematics as the GAMMA function; see http://www.wikipedia.org/wiki/Gamma_function for a description. Also, see *Statistical Theory*, 3rd Ed., B.W. Lindgren, MacMillian Pub. Inc, 1976, pp. 336-337.

[2] Computer software packages such as SAS and EXCEL have implemented statistical and mathematical functions to readily calculate values for T_Dist{ . } and TINV_Dist{ . } as defined above.

Table 6.1

YEAR	MINAVG	Month(MinAvg)
1950	40.86	Jan
1951	44.57	Dec
1952	43.07	Jan
1953	45.69	Feb
1954	45.70	Dec
1955	45.84	Dec
1956	44.91	Feb
1957	39.50	Jan
1958	46.27	Nov
1959	48.26	Feb
1960	42.33	Jan
1961	47.22	Dec
1962	43.42	Jan
1963	42.61	Jan
1964	45.24	Nov
1965	44.80	Jan
1966	46.72	Jan
1967	40.77	Dec
1968	40.64	Dec
1969	44.85	Jan
1970	46.83	Dec
1971	43.01	Jan
1972	41.43	Dec
1973	45.07	Jan
1974	42.99	Jan
1975	44.64	Jan
1976	44.85	Jan
1977	48.35	Jan
1978	41.66	Dec
1979	41.39	Jan
1980	50.36	Jan
1981	49.34	Jan
1982	45.35	Jan
1983	48.69	Jan
1984	46.92	Dec
1985	45.13	Feb
1986	48.60	Feb
1987	43.46	Dec
1988	43.29	Dec
1989	40.61	Feb
1990	39.01	Dec
1991	48.68	Mar
1992	47.36	Dec
1993	46.12	Jan
1994	47.16	Nov

Table 6.2

YEAR	MINAVG	Month(MinAvg)
1995	49.85	Dec
1996	44.96	Feb
1997	48.38	Jan
1998	43.64	Dec
1999	49.01	Jan
2000	48.79	Mar
2001	47.17	Feb
2002	45.82	Jan
2003	47.09	Dec
2004	48.22	Nov
2005	47.28	Jan
2006	45.80	Mar
2007	41.54	Jan
2008	45.81	Dec
2009	45.27	Dec
2010	44.71	Dec
2011	46.76	Feb
2012	46.78	Dec
2013	43.92	Jan
2014	48.07	Dec
2015	45.62	Jan
2016	46.74	Dec
2017	47.58	Jan
2018	47.38	Feb
2019	47.27	Feb
2020	50.00	Feb
2021	46.98	Jan

Table 7.1

alpha= 0.375

Year	Days/Yr	-MinAvg	Month (-MinAvg)	Rank	Emprical CDF	Fitted Model CDF
1980	366	-50.3562	Jan	1	0.0087	-1.8315
2020	366	-49.9988	Feb	2	0.0225	-1.6975
1995	365	-49.8502	Dec	3	0.0363	-1.6418
1981	365	-49.3410	Jan	4	0.0502	-1.4509
1999	365	-49.0143	Jan	5	0.0640	-1.3284
2000	366	-48.7946	Mar	6	0.0779	-1.2460
1983	365	-48.6916	Jan	7	0.0917	-1.2074
1991	365	-48.6770	Mar	8	0.1055	-1.2019
1986	365	-48.5968	Feb	9	0.1194	-1.1718
1997	365	-48.3795	Jan	10	0.1332	-1.0903
1977	365	-48.3454	Jan	11	0.1471	-1.0775
1959	365	-48.2581	Feb	12	0.1609	-1.0448
2004	366	-48.2200	Nov	13	0.1747	-1.0305
2014	365	-48.0744	Dec	14	0.1886	-0.9759
2017	365	-47.5793	Jan	15	0.2024	-0.7902
2018	365	-47.3752	Feb	16	0.2163	-0.7137
1992	366	-47.3557	Dec	17	0.2301	-0.7064
2005	365	-47.2788	Jan	18	0.2439	-0.6776
2019	365	-47.2671	Feb	19	0.2578	-0.6732
1961	365	-47.2162	Dec	20	0.2716	-0.6541
2001	365	-47.1654	Feb	21	0.2855	-0.6350
1994	365	-47.1570	Nov	22	0.2993	-0.6319
2003	365	-47.0899	Dec	23	0.3131	-0.6068
2021	365	-46.9762	Jan	24	0.3270	-0.5641
1984	366	-46.9228	Dec	25	0.3408	-0.5441
1970	365	-46.8300	Dec	26	0.3547	-0.5093
2012	366	-46.7772	Dec	27	0.3685	-0.4895
2011	365	-46.7623	Feb	28	0.3824	-0.4839
2016	366	-46.7389	Dec	29	0.3962	-0.4751
1966	365	-46.7161	Jan	30	0.4100	-0.4666
1958	365	-46.2675	Nov	31	0.4239	-0.2984
1993	365	-46.1152	Jan	32	0.4377	-0.2412
1955	365	-45.8398	Dec	33	0.4516	-0.1380
2002	365	-45.8224	Jan	34	0.4654	-0.1315
2008	366	-45.8121	Dec	35	0.4792	-0.1276
2006	365	-45.8025	Mar	36	0.4931	-0.1240
1954	365	-45.6962	Dec	37	0.5069	-0.0841
1953	365	-45.6852	Feb	38	0.5208	-0.0800
2015	365	-45.6211	Jan	39	0.5346	-0.0560
1982	365	-45.3516	Jan	40	0.5484	0.0451
2009	365	-45.2689	Dec	41	0.5623	0.0761
1964	366	-45.2362	Nov	42	0.5761	0.0884
1985	365	-45.1295	Feb	43	0.5900	0.1284
1973	365	-45.0719	Jan	44	0.6038	0.1500
1996	366	-44.9572	Feb	45	0.6176	0.1930

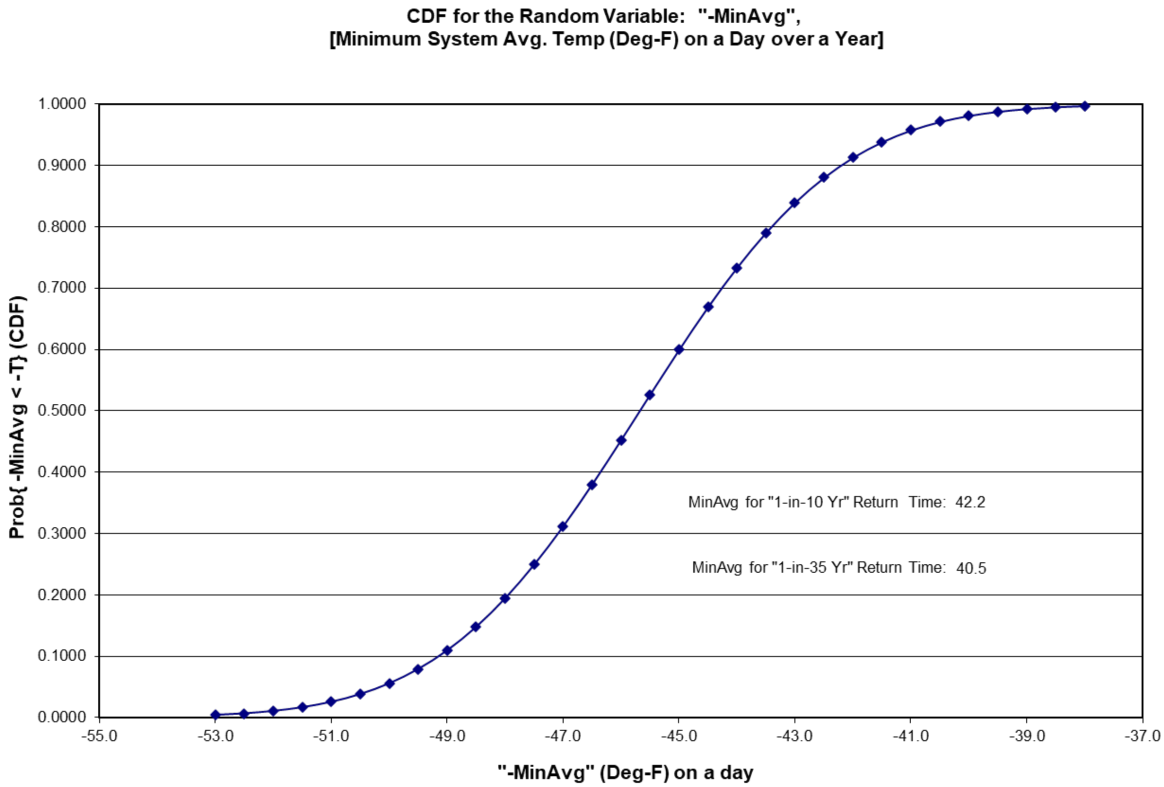
Table 7.2

alpha= 0.375

Year	Days/Yr	-MinAvg	Month (-MinAvg)	Rank	Emprical CDF	Fitted Model CDF
1956	366	-44.9092	Feb	46	0.6315	0.2110
1976	366	-44.8492	Jan	47	0.6453	0.2335
1969	365	-44.8451	Jan	48	0.6592	0.2350
1965	365	-44.8016	Jan	49	0.6730	0.2513
2010	365	-44.7107	Dec	50	0.6869	0.2854
1975	365	-44.6416	Jan	51	0.7007	0.3113
1951	365	-44.5690	Dec	52	0.7145	0.3386
2013	365	-43.9179	Jan	53	0.7284	0.5827
1998	365	-43.6433	Dec	54	0.7422	0.6857
1987	365	-43.4643	Dec	55	0.7561	0.7528
1962	365	-43.4218	Jan	56	0.7699	0.7687
1988	366	-43.2901	Dec	57	0.7837	0.8181
1952	366	-43.0724	Jan	58	0.7976	0.8997
1971	365	-43.0081	Jan	59	0.8114	0.9239
1974	365	-42.9915	Jan	60	0.8253	0.9301
1963	365	-42.6117	Jan	61	0.8391	1.0725
1960	366	-42.3298	Jan	62	0.8529	1.1782
1978	365	-41.6636	Dec	63	0.8668	1.4281
2007	365	-41.5391	Jan	64	0.8806	1.4747
1972	366	-41.4268	Dec	65	0.8945	1.5168
1979	365	-41.3863	Jan	66	0.9083	1.5320
1950	365	-40.8618	Jan	67	0.9221	1.7287
1967	365	-40.7720	Dec	68	0.9360	1.7624
1968	366	-40.6420	Dec	69	0.9498	1.8111
1989	365	-40.6067	Feb	70	0.9637	1.8244
1957	365	-39.5002	Jan	71	0.9775	2.2393
1990	365	-39.0145	Dec	72	0.9913	2.4214

"Gamma" (Fitted) =	**-45.68**
"Theta" (Fitted) =	**2.68**
Deg. Freedom=	**70**

Figure 1


CDF for the Random Variable: "-MinAvg",
[Minimum System Avg. Temp (Deg-F) on a Day over a Year]
Prob{ -MinAvg < -T} (CDF)
1.0000
0.9000
0.8000
0.7000
0.6000
0.5000
0.4000
0.3000
0.2000
0.1000
0.0000
MinAvg for "1-in-10 Yr" Return Time: 42.2
MinAvg for "1-in-35 Yr" Return Time: 40.5
-55.0
-53.0
-51.0
-49.0
-47.0
-45.0
-43.0
-41.0
-39.0
-37.0
"-MinAvg" (Deg-F) on a day

VI. Estimating the Uncertainty in the Peak-Day Design Temperature

The calculated peak-day design temperatures in section V above also have a statistical uncertainty associated with them. The estimated measures of uncertainty recommended for our use are calculated from the fitted model for the probability distribution and are believed to be reasonable, although rough, approximations.

The basic approach used the estimated parameters for the probability distribution (see the results provided in Tables 7.1 and 7.2, above) to calculate the fitted temperatures as a function of the empirical CDF listed in Tables 7.1 and 7.2, above. These fitted temperatures are then compared with the observed temperatures by calculating the difference = "observed" – "fitted" values. The full set of differences are then separated into the lower third (L), the middle third (M) and the upper third (U) of the distribution. Finally, values of the root-mean-square error (RMSE) of the differences in each third of the distribution are calculated, along with the RMSE for the entire set of differences overall. The data in Tables 8.1 and 8.2, below, show the temperature data and the resulting RMSE values.

The formula below is used to calculate the RMSE for a specified set of "N" data differences:

$$\text{RMSE} = \text{SQRT}\left\{ \left(\sum_{i=1, \ldots, N} e[\, i\,]^{2} \right) / \, (N-2) \right\},$$

where e[i] = *observed* less *fitted* value of temperature, T[i]. The number of estimated parameters (3 for the GEV model, 2 for the T-Dist and EV1 models) is subtracted from the respective number of data differences, N, in the denominator of the RMSE expression.

Since both the "1-in-35" and "1-in-10" peak-day temperature values are in the lower third quantile of the fitted distribution, the calculated standard error for these estimates is 0.57 Deg-F.

Table 8.1

Quantile: (Lower, Middle, Upper 3rd's)	Observed $T_{[i]}$ Temp. Ranked	Fitted Value of $T_{[i]}$	Residual $e_{[i]}$: Obs'd. less Fitted Value of $T_{[i]}$	Square of $e_{[i]}$:
U	50.3562	52.2131	-1.8569	3.448046
U	49.9988	51.1494	-1.1506	1.323977
U	49.8502	50.5625	-0.7123	0.507337
U	49.3410	50.1411	-0.8001	0.640203
U	49.0143	49.8060	-0.7917	0.626760
U	48.7946	49.5244	-0.7299	0.532690
U	48.6916	49.2794	-0.5878	0.345459
U	48.6770	49.0609	-0.3840	0.147427
U	48.5968	48.8627	-0.2659	0.070724
U	48.3795	48.6805	-0.3010	0.090595
U	48.3454	48.5111	-0.1657	0.027463
U	48.2581	48.3524	-0.0942	0.008882
U	48.2200	48.2025	0.0175	0.000307
U	48.0744	48.0601	0.0143	0.000203
U	47.5793	47.9242	-0.3449	0.118968
U	47.3752	47.7938	-0.4186	0.175220
U	47.3557	47.6683	-0.3126	0.097689
U	47.2788	47.5470	-0.2682	0.071946
U	47.2671	47.4295	-0.1624	0.026360
U	47.2162	47.3153	-0.0990	0.009810
U	47.1654	47.2040	-0.0386	0.001490
U	47.1570	47.0952	0.0618	0.003817
U	47.0899	46.9888	0.1011	0.010220
U	46.9762	46.8845	0.0917	0.008403
M	46.9228	46.7819	0.1409	0.019847
M	46.8300	46.6810	0.1489	0.022180
M	46.7772	46.5815	0.1957	0.038290
M	46.7623	46.4833	0.2791	0.077888
M	46.7389	46.3861	0.3528	0.124477
M	46.7161	46.2898	0.4262	0.181666
M	46.2675	46.1944	0.0731	0.005343
M	46.1152	46.0996	0.0156	0.000242
M	45.8398	46.0053	-0.1656	0.027411
M	45.8224	45.9115	-0.0891	0.007933
M	45.8121	45.8179	-0.0058	0.000034
M	45.8025	45.7246	0.0779	0.006072
M	45.6962	45.6312	0.0650	0.004227
M	45.6852	45.5378	0.1474	0.021713
M	45.6211	45.4443	0.1769	0.031276
M	45.3516	45.3504	0.0011	0.000001
M	45.2689	45.2562	0.0128	0.000163
M	45.2362	45.1614	0.0748	0.005595
M	45.1295	45.0659	0.0636	0.004044
M	45.0719	44.9697	0.1022	0.010452
M	44.9572	44.8725	0.0846	0.007162
M	44.9092	44.7743	0.1350	0.018217
M	44.8492	44.6748	0.1745	0.030433
M	44.8451	44.5738	0.2713	0.073580

Table 8.2

Quantile: (Lower, Middle, Upper 3rd's)	Observed $T_{[i]}$ Temp. Ranked	Fitted Value of $T_{[i]}$	Residual $e_{[i]}$: Obs'd. less Fitted Value of $T_{[i]}$	Square of $e_{[i]}$:
L	44.8016	44.4713	0.3303	0.109112
L	44.7107	44.3670	0.3438	0.118166
L	44.6416	44.2605	0.3811	0.145228
L	44.5690	44.1518	0.4172	0.174035
L	43.9179	44.0405	-0.1227	0.015045
L	43.6433	43.9263	-0.2830	0.080089
L	43.4643	43.8088	-0.3445	0.118689
L	43.4218	43.6875	-0.2657	0.070582
L	43.2901	43.5620	-0.2718	0.073897
L	43.0724	43.4316	-0.3592	0.129007
L	43.0081	43.2957	-0.2876	0.082718
L	42.9915	43.1533	-0.1618	0.026174
L	42.6117	43.0034	-0.3917	0.153447
L	42.3298	42.8447	-0.5149	0.265102
L	41.6636	42.6753	-1.0117	1.023556
L	41.5391	42.4931	-0.9540	0.910045
L	41.4268	42.2949	-0.8681	0.753526
L	41.3863	42.0764	-0.6901	0.476274
L	40.8618	41.8314	-0.9696	0.940056
L	40.7720	41.5498	-0.7778	0.604952
L	40.6420	41.2147	-0.5727	0.328023
L	40.6067	40.7933	-0.1866	0.034824
L	39.5002	40.2063	-0.7061	0.498579
L	39.0145	39.1427	-0.1283	0.016450

Overall RMSE ($e_{[i]}$):	**0.48**	**°F**
Upper 3rd RMSE ($e_{[i]}$):	**0.61**	**°F**
Middle 3rd RMSE ($e_{[i]}$):	**0.18**	**°F**
Lower 3rd RMSE ($e_{[i]}$):	**0.57**	**°F**

VII. The Relationship between Annual Likelihoods for Peak-Day Temperatures and "Expected Return Time"

The event whose probability distribution we've modeled is the likelihood that the minimum daily temperature over a calendar year is less than a specified value. And, in particular, we've used this probability model to infer the value of a temperature, our *peak-day design temperature* ($TPDD_\delta$), that corresponds to a pre-defined likelihood, δ, that the observed minimum temperature is less than or equal to this design temperature.

(1) δ = Prob{ Minimum Daily Temperature over the Year < $TPDD_\delta$ }.

For some applications, it is useful to think of how this specified likelihood (or "risk level" δ) relates to the expected number of years until this Peak-Day event would first occur. This expected number of years is what is meant by the *return period*. The results stated below are found in the book: ***Statistics of Extremes***, E.J. Gumbel, Columbia University Press, 1958, on pages 21-25.

(2) E[#Yrs for Peak-Day Event to Occur] = 1 / δ,

1 / Prob{ Minimum Daily Temperature over the Year < $TPDD_\delta$ }.

For our peak-day design temperature (40.5°F) associated with a 1-in-35 annual likelihood, the return period is 35 years (δ=1/35). For the 42.2°F peak-day design temperature, the return period is 10 years (δ=1/10). Occasionally, a less precise terminology is used. For example, the 40.5°F peak-day design temperature may be referred to as a "1-in-35 year cold day"; and the 42.2°F peak-day design temperature may be referred to as a "1-in-10 year cold day."

The probability model for the *return period*, as a random variable, is a geometric (discrete) distribution with positive integer values for the *return period*. The parameter δ = Prob{ Minimum Daily Temperature over the Year < $TPDD_\delta$ }.

(3) Prob{ *return period* = r } = $(1 - \delta)^{(r-1)} \delta$, for r = 1, 2, 3, . . .

The expected value of the *return period* is already given in (2) above; the variance of the *return period* is:

(4) Var[*return period*] = (E[*return period*])^2 x (1- (1 / E [*return period*])),

(4') Var[*return period*] = (E[*return period*]) x (E [*return period*] - 1).

Equations (4) and (4') indicate that the standard deviation (square root of the variance) of the *return period* is nearly equal to its expected value. Thus, there is substantial variability about the expected value—a *return period* is not very precise.

SoCalGas SAS Code #1: SysAvgVar(Daily).sas

```
Title1 "Combine PRELIMINARY weather data (from 'wdaily46.sas7bdat') and combine with previous" ;
Title2 "FINAL NWS data ('dly50_96.sas7bdat') and UPDATE ('Updt4NWS.sas7bdat') from TWO NWS CD-ROM files. " ;

/*********************************************************************************/
/*                                                                               */
/* Two SAS-Formated data sets are created:                                       */
/*    1). "scgwea.DlySys_D" contains Daily System Average values of HDD, CDD     */
/*        and Temperature ("AVG")                                                */
/*    2). "scgwea.DlyStn_D" contains Daily of HDD,CDD and Temperature            */
/*        for each of our 15 weather stations, our 6-Climate Zones, and our      */
/*        "System Averages"                                                      */
/*                                                                               */
/*********************************************************************************/




options mprint ;
options ls=160 ps=90 ; **<<PORTRAIT:   SAS-Monospace w/Roman 6pt. Font >>**;
options date number notes ;
/* %cour8l
   %cour8p */
*options ls=211 ps=69 ; **<<LANDSCAPE:  SAS-Monospace w/Roman 6pt. Font >>**;



*********************************************************************************;

* Data Importation Section;

*libname scgdf '/data/home/scgdf';
libname scgwea '/EDS_RB/Weather/2022 CGR/SoCalGas';

**** Calculate Hdd, Cdd from PRELIMINARY data set! ****;
****libname InWea  '/data/home/scgdf/weather/wdaily46.sas7bdat';

  **<< Use the copy of "wdaily46.sas7bdat" saved at this folder. >>** ;
  **<< It has data we need through Mar 13, 2022 for both SoCalGas and the SDGandE locations. >>** ;



proc contents data=scgwea.wdaily46 ;  **<< File was current through Mar 13, 2022. >>>*** ;
run ;


**<< New Weights for 6-Climate Zones: (per Avg. Mtr. Cnt for Yr)        >>** ;

**<<   Zone:       <<Dec 2019 Values>>  <<Year 20xx Values>>   >>** ;
**<< 1 High Mtn.      0.005632            a.aaaa          >>** ;
**<< 2 Low Desert     0.037582            b.bbbb          >>** ;
**<< 3 Coastal        0.179423            c.cccc          >>** ;
**<< 4 High Desert    0.070095            d.dddd          >>** ;
**<< 5 Intr. Valleys  0.373126            e.eeee          >>** ;
**<< 6 Basin          0.334142            f.ffff          >>** ;
**<< -------------- ---------             ------          >>** ;
**<<   TOTAL          1.000000            1.0000          >>** ;
```

```
**<<   Zone:       <<Dec 2017 Values>>  <<Year 20xx Values>>   >>** ;
**<< 1 High Mtn.     0.005822                 a.aaaa          >>** ;
**<< 2 Low Desert    0.038497                 b.bbbb          >>** ;
**<< 3 Coastal       0.185384                 c.cccc          >>** ;
**<< 4 High Desert   0.071615                 d.dddd          >>** ;
**<< 5 Intr. Valleys 0.383107                 e.eeee          >>** ;
**<< 6 Basin         0.315576                 f.ffff          >>** ;
**<< -------------- ---------                 ------          >>** ;
**<<   TOTAL         1.000000                 1.0000          >>** ;


**<<   Zone:       <<Dec 2015 Values>>  <<Year 20xx Values>>   >>** ;
**<< 1 High Mtn.     0.005842                 a.aaaa          >>** ;
**<< 2 Low Desert    0.038341                 b.bbbb          >>** ;
**<< 3 Coastal       0.186160                 c.cccc          >>** ;
**<< 4 High Desert   0.071125                 d.dddd          >>** ;
**<< 5 Intr. Valleys 0.385713                 e.eeee          >>** ;
**<< 6 Basin         0.312819                 f.ffff          >>** ;
**<< -------------- ---------                 ------          >>** ;
**<<   TOTAL         1.000000                 1.0000          >>** ;


**<<   Zone:       <<Dec 2014 Values>>  <<Year 20xx Values>>   >>** ;
**<< wtz5 calcd value was 0.3813, but used 0.3812 so total adds to 100.00 exactly. >>** ;
**<< 1 High Mtn.     0.0061            a.aaaa          >>** ;
**<< 2 Low Desert    0.0423            b.bbbb          >>** ;
**<< 3 Coastal       0.1763            c.cccc          >>** ;
**<< 4 High Desert   0.0747            d.dddd          >>** ;
**<< 5 Intr. Valleys 0.3812  (0.3813)      e.eeee          >>** ;
**<< 6 Basin         0.3194            f.ffff          >>** ;
**<< -------------- -------            -------         >>** ;
**<<   TOTAL         1.0000            1.0000          >>** ;


**<<   Zone:       <<Dec 2013 Values>>  <<Year 20xx Values>>   >>** ;
**<< wtz5 calcd value was 0.3793, but used 0.3792 so total adds to 100.00 exactly. >>** ;
**<< 1 High Mtn.     0.0062            a.aaaa          >>** ;
**<< 2 Low Desert    0.0424            b.bbbb          >>** ;
**<< 3 Coastal       0.1772            c.cccc          >>** ;
**<< 4 High Desert   0.0742            d.dddd          >>** ;
**<< 5 Intr. Valleys 0.3792  (0.3793)      e.eeee          >>** ;
**<< 6 Basin         0.3208            f.ffff          >>** ;
**<< -------------- -------            -------         >>** ;
**<<   TOTAL         1.0000            1.0000          >>** ;


**<< New Weights for 6-Climate Zones: (per Avg. Mtr. Cnt for Yr)        >>** ;

**<<   Zone:       <<Dec 2010 Values>>  <<Year 20xx Values>>   >>** ;
**<< wtz6 calcd value was 0.3199, but used 0.3198 so total adds to 100.00 exactly. >>** ;
**<< 1 High Mtn.     0.0062            a.aaaa          >>** ;
**<< 2 Low Desert    0.0418            b.bbbb          >>** ;
**<< 3 Coastal       0.1774            c.cccc          >>** ;
**<< 4 High Desert   0.0746            d.dddd          >>** ;
**<< 5 Intr. Valleys 0.3802            e.eeee          >>** ;
**<< 6 Basin         0.3198            f.ffff          >>** ;
**<< -------------- -------            -------         >>** ;
**<<   TOTAL         1.0000            1.0000          >>** ;

**<< New Weights for 6-Climate Zones: (per Avg. Mtr. Cnt for Yr)        >>** ;
```

```
**<<   Zone:       <<Dec 2009 Values>>  <<Year 20xx Values>>   >>** ;
**<< 1 High Mtn.     0.0062              a.aaaa          >>** ;
**<< 2 Low Desert    0.0417              b.bbbb          >>** ;
**<< 3 Coastal       0.1779              c.cccc          >>** ;
**<< 4 High Desert   0.0745              d.dddd          >>** ;
**<< 5 Intr. Valleys 0.3794              e.eeee          >>** ;
**<< 6 Basin         0.3203              f.ffff          >>** ;
**<< -------------- -------              -------         >>** ;
**<<   TOTAL         1.0000              1.0000          >>** ;

**<< New Weights for 6-Climate Zones: (per Avg. Mtr. Cnt for Yr)        >>** ;

**<<   Zone:       <<Year 2006 Values>>  <<Year 20xx Values>>   >>** ;
**<< 1 High Mtn.     0.0057              a.aaaa          >>** ;
**<< 2 Low Desert    0.0354              b.bbbb          >>** ;
**<< 3 Coastal       0.1888              c.cccc          >>** ;
**<< 4 High Desert   0.0676              d.dddd          >>** ;
**<< 5 Intr. Valleys 0.3854              e.eeee          >>** ;
**<< 6 Basin         0.3171              f.ffff          >>** ;
**<< -------------- -------              -------         >>** ;
**<<   TOTAL         1.0000              1.0000          >>** ;

**<<   Zone:       <<Year 2002 Values>>  <<Year 20xx Values>>   >>** ;
**<< 1 High Mtn.     0.0062              a.aaaa          >>** ;
**<< 2 Low Desert    0.0332              b.bbbb          >>** ;
**<< 3 Coastal       0.1998              c.cccc          >>** ;
**<< 4 High Desert   0.0662              d.dddd          >>** ;
**<< 5 Intr. Valleys 0.3807              e.eeee          >>** ;
**<< 6 Basin         0.3139              f.ffff          >>** ;
**<< -------------- -------              -------         >>** ;
**<<   TOTAL         1.0000              1.0000          >>** ;

**<<   Zone:      Year 1992 values    <<Year 2000 Values>>  <<Year 2001 Values>>   >>** ;
**<< 1 High Mtn.     0.0065              0.0063              0.0062          >>** ;
**<< 2 Low Desert    0.0282              0.0312              0.0320          >>** ;
**<< 3 Coastal       0.1900              0.2020              0.2021          >>** ;
**<< 4 High Desert   0.0620              0.0655              0.0638          >>** ;
**<< 5 Intr. Valleys 0.3862              0.3799              0.3809          >>** ;
**<< 6 Basin         0.3271              0.3151              0.3150          >>** ;
**<< -------------- -------              -------             -------         >>** ;
**<<   TOTAL         1.0000              1.0000              1.0000          >>** ;


%global wtz1 wtz2 wtz3 wtz4 wtz5 wtz6 ;

**<< Zone Weights a/o December, 2021.      >>** ;
**<< (Data kindly provided by Andrew Tung.) >>** ;
%let wtz1=0.005730 ; **<UPDATE>;
%let wtz2=0.038600 ; **<UPDATE>;
%let wtz3=0.182121 ; **<UPDATE>;
%let wtz4=0.072204 ; **<UPDATE>;
%let wtz5=0.381874 ; **<UPDATE>;
%let wtz6=0.319470 ; **<UPDATE>;



**<< Code inserted to do the daily "system avg." calc. from Six-Zone formulas  >>**;
**<< that are used to calc. the daily "system avg." values for the NWS data.   >>**;

data prelimD ;
   set scgwea.wdaily46(keep=date
```

```
          max_1-max_46
          min_1-min_46 );

  ** if date >= '01jan97'd; **<< Previous NWS data w/update goes through Dec. 31, 2000.    >>** ;

  ** if date >= '01nov02'd; **<< Current NWS data w/update NOW goes through Oct. 31, 2002. >>**;

    if date >= mdy(11,01,2002); **<< Current NWS data w/update NOW goes through Oct. 31, 2002. >>**;
                              **<< Use this format to "test" the DATE variable!!            >>**;

    year = year(date);
    month = month(date);

    ARRAY A_MAX(i) S0442MAX S1048MAX S1194MAX S4671MAX
                              S5085MAX S5114MAX S5115MAX S6175MAX
                              S6624MAX S6635MAX S6719MAX S7050MAX
                              S7306MAX S7851MAX S7888MAX S7905MAX S9367MAX;

          ARRAY A_MIN(i) S0442MIN S1048MIN S1194MIN S4671MIN
                              S5085MIN S5114MIN S5115MIN S6175MIN
                              S6624MIN S6635MIN S6719MIN S7050MIN
                              S7306MIN S7851MIN S7888MIN S7905MIN S9367MIN;

          ARRAY A_AVG(i) S0442AVG S1048AVG S1194AVG S4671AVG
                              S5085AVG S5114AVG S5115AVG S6175AVG
                              S6624AVG S6635AVG S6719AVG S7050AVG
                              S7306AVG S7851AVG S7888AVG S7905AVG S9367AVG;

          ARRAY A_hdd(i) S0442hdd S1048hdd S1194hdd S4671hdd
                              S5085hdd S5114hdd S5115hdd S6175hdd
                              S6624hdd S6635hdd S6719hdd S7050hdd
                              S7306hdd S7851hdd S7888hdd S7905hdd S9367hdd;

          ARRAY A_cdd(i) S0442cdd S1048cdd S1194cdd S4671cdd
                              S5085cdd S5114cdd S5115cdd S6175cdd
                              S6624cdd S6635cdd S6719cdd S7050cdd
                              S7306cdd S7851cdd S7888cdd S7905cdd S9367cdd;

  ** array a_max(i) max_1-max_46;  **<< There are NOW 46 stations on this file! >>**;
  ** array a_min(i) min_1-min_46;
  ** array a_avg(i) avg_1-avg_46;
  ** array a_hdd(i) hdd_1-hdd_46;
  ** array a_cdd(i) cdd_1-cdd_46;

 %macro MapStatn(el);
 *****************    "NWS" Data-Station        <---  "WeaDaily" Prelim Data-Station ***;
   s5115&el = (&el._3 ) ; *  LA Civic Cntr.     <---  Los Angeles Civic Center         ;
   s5114&el = (&el._4 ) ; *  LA Air Port        <---  Los Angeles International Apt.   ;
   s1194&el = (&el._9 ) ; *  Burbank            <---  Burbank                          ;
   s7050&el = (&el._11) ; *  Pomona             <---  Ontario                          ;
   s6719&el = (&el._12) ; *  Pasadena           <---  Pasadena                         ;
   s7306&el = (&el._46) ; *  Redlands           <---  RIALTO (replaced San Bernardino, Aug-2005)  ;
   s6175&el = (&el._20) ; *  Newport Beach      <---  Newport Beach                    ;
   s7888&el = (&el._21) ; *  Santa Ana          <---  Santa Ana                        ;
   s4671&el = (&el._23) ; *  Lake Arrowhead     <---  Big Bear                         ;
   s6624&el = (&el._24) ; *  Palmdale           <---  Lancaster                        ;
   s1048&el = (&el._26) ; *  Brawley            <---  El Centro                        ;
   s6635&el = (&el._28) ; *  Palm Springs       <---  Palm Springs                     ;
   s7905&el = (&el._31) ; *  Santa Barbara-Apt  <---  Santa Barbara (City?)            ;
   s0442&el = (&el._34) ; *  Bakersfield        <---  Bakersfield                      ;
   s9367&el = (&el._35) ; *  Visalia            <---  Fresno                           ;
```

```
  **<< Note: The two NWS stations below are NOT used in the climate zone averages. >>**;
  **<<       The NWS data set("dly50_96") has 17 stations, only 15 are used.       >>**;
  s5085&el = (&el._2 ) ; *  Long Beach(Apt)      <---  Long Beach                     ;
  s7851&el = (&el._32) ; *  San Luis Obispo(@CSU)<---  San Luis Obispo                ;

%mend;


%mapstatn(max);
%mapstatn(min);

*** %mapstatn(avg); **<< Do not need to do these calcs.! >>** ;
*** %mapstatn(hdd);
*** %mapstatn(cdd);

    do over a_max;
      a_avg = (a_max+a_min)/2;
      if a_avg-int(a_avg)=.5 then if mod(int(a_avg),2)=0
      then a_avg=int(a_avg);
      else a_avg=int(a_avg)+1;
      a_hdd = max(0,65-a_avg);
      a_cdd = max(0,a_avg-65);
    end;



  **<< Calc. Climate Zones and "System Average" values for:          >>**;
  **<<          Hdd, Cdd and Avg (temperature)                       >>**;

    do over a_hdd;
      a_hdd = round(a_hdd);
      a_cdd = round(a_cdd);
    end;


    hddz1 =   S4671hdd * &wtz1;
    hddz2 = ((S1048hdd + S6635hdd)/2) * &wtz2;
    hddz3 = ((S5114hdd + S6175hdd + S7905hdd)/3) * &wtz3;
    hddz4 = ((S0442hdd + S6624hdd + S9367hdd)/3) * &wtz4;
    hddz5 = ((S1194hdd + S6719hdd + S7050hdd + S7306hdd)/4) * &wtz5;
    hddz6 = ((S7888hdd + S5115hdd)/2) * &wtz6;
  **hdd   = round(sum (of hddz1-hddz6),0.1);
    hdd   =       sum (of hddz1-hddz6)     ;  **<< Do not round! >>** ;

    cddz1 =   S4671cdd * &wtz1;
    cddz2 = ((S1048cdd + S6635cdd)/2) * &wtz2;
    cddz3 = ((S5114cdd + S6175cdd + S7905cdd)/3) * &wtz3;
    cddz4 = ((S0442cdd + S6624cdd + S9367cdd)/3) * &wtz4;
    cddz5 = ((S1194cdd + S6719cdd + S7050cdd + S7306cdd)/4) * &wtz5;
    cddz6 = ((S7888cdd + S5115cdd)/2) * &wtz6;
  **cdd   = round(sum (of cddz1-cddz6),0.1);
    cdd   =       sum (of cddz1-cddz6)     ;  **<< Do not round! >>** ;

    avgz1 =   S4671avg * &wtz1;
    avgz2 = ((S1048avg + S6635avg)/2) * &wtz2;
    avgz3 = ((S5114avg + S6175avg + S7905avg)/3) * &wtz3;
    avgz4 = ((S0442avg + S6624avg + S9367avg)/3) * &wtz4;
    avgz5 = ((S1194avg + S6719avg + S7050avg + S7306avg)/4) * &wtz5;
    avgz6 = ((S7888avg + S5115avg)/2) * &wtz6;
  **avg   = round(sum (of avgz1-avgz6),0.1);
    avg   =       sum (of avgz1-avgz6)     ;  **<< Do not round! >>** ;
```

```
     drop   max_1-max_46
            min_1-min_46
            avg_1-avg_46
            hdd_1-hdd_46
            cdd_1-cdd_46;     **<< Drop ALL the PRELIMINARY weather data! >>** ;

run;




**** Calculate  HDD and CDD from FINAL NWS data set (dly50_96.sas7bdat) w/update
(Updt4NWS.sas7bdat). ****;

* Commented Out by GDT;
***libname in 'S:\Weather\2005Bcap\Updt-NWS\';
***libname in2 'S:\Weather\2005Bcap\Updt-NWS\';

* Commented Out by GDT;
*libname in 'C:\Weather\2005Bcap\Updt-NWS\';
*libname in2 'C:\Weather\2005Bcap\Updt-NWS\';

  **<< This is the built-in drive on the computer BMW uses when telecommuting.  >>**;
  **<< It also has a copy of the same "wdaily46.sd2" file on the "S:\" drive.   >>**;
  **<< Having the file on the built-in drive allows for faster execution.       >>**;



 proc contents data=scgwea.dly50_96 ;
 run ;


 proc contents data=scgwea.Updt4NWS ;
 run ;



/*****************************
*****************************/


data Updt4NWS ;
     set scgwea.Updt4NWS ;

  **<< Keep only the Post-1996 observations thorugh YE-2002.              >>** ;
  **<< Note: the file "in2.Updt4NWS" actually goes through Oct. 31st, 2002! >>** ;
  if ( (date > mdy(12,31,1996))
   and (date < mdy(11,01,2002)) ) ;

run ;


data finalD ;
     merge scgwea.dly50_96  Updt4NWS ;
   by date ;

     year  = year(date);
```

```
   month = month(date);
   ARRAY A_MAX(I) S0442MAX S1048MAX S1194MAX S4671MAX
          S5085MAX S5114MAX S5115MAX S6175MAX
          S6624MAX S6635MAX S6719MAX S7050MAX
          S7306MAX S7851MAX S7888MAX S7905MAX S9367MAX;
   ARRAY A_MIN(I) S0442MIN S1048MIN S1194MIN S4671MIN
          S5085MIN S5114MIN S5115MIN S6175MIN
          S6624MIN S6635MIN S6719MIN S7050MIN
          S7306MIN S7851MIN S7888MIN S7905MIN S9367MIN;
   ARRAY A_AVG(I) S0442AVG S1048AVG S1194AVG S4671AVG
          S5085AVG S5114AVG S5115AVG S6175AVG
          S6624AVG S6635AVG S6719AVG S7050AVG
          S7306AVG S7851AVG S7888AVG S7905AVG S9367AVG;
   ARRAY A_hdd(I) S0442hdd S1048hdd S1194hdd S4671hdd
          S5085hdd S5114hdd S5115hdd S6175hdd
          S6624hdd S6635hdd S6719hdd S7050hdd
          S7306hdd S7851hdd S7888hdd S7905hdd S9367hdd;
   ARRAY A_cdd(I) S0442cdd S1048cdd S1194cdd S4671cdd
          S5085cdd S5114cdd S5115cdd S6175cdd
          S6624cdd S6635cdd S6719cdd S7050cdd
          S7306cdd S7851cdd S7888cdd S7905cdd S9367cdd;
   DO OVER A_MAX;
     A_AVG = (A_MAX+A_MIN)/2;
    IF A_AVG-INT(A_AVG)=.5 THEN IF MOD(INT(A_AVG),2)=0
    THEN A_AVG=INT(A_AVG);
    ELSE A_AVG=INT(A_AVG)+1;
    a_hdd = max(0,65-a_avg);
    a_cdd = max(0,a_avg-65);
   END;


   hddz1 =   S4671hdd * &wtz1;
   hddz2 = ((S1048hdd + S6635hdd)/2) * &wtz2;
   hddz3 = ((S5114hdd + S6175hdd + S7905hdd)/3) * &wtz3;
   hddz4 = ((S0442hdd + S6624hdd + S9367hdd)/3) * &wtz4;
   hddz5 = ((S1194hdd + S6719hdd + S7050hdd + S7306hdd)/4) * &wtz5;
   hddz6 = ((S7888hdd + S5115hdd)/2) * &wtz6;
 ** hdd   = round(sum (of hddz1-hddz6),0.1);
   hdd   =      sum (of hddz1-hddz6)     ;   **<< Do not round! >>** ;

   cddz1 =   S4671cdd * &wtz1;
   cddz2 = ((S1048cdd + S6635cdd)/2) * &wtz2;
   cddz3 = ((S5114cdd + S6175cdd + S7905cdd)/3) * &wtz3;
   cddz4 = ((S0442cdd + S6624cdd + S9367cdd)/3) * &wtz4;
   cddz5 = ((S1194cdd + S6719cdd + S7050cdd + S7306cdd)/4) * &wtz5;
   cddz6 = ((S7888cdd + S5115cdd)/2) * &wtz6;
 ** cdd   = round(sum (of cddz1-cddz6),0.1);
   cdd   =      sum (of cddz1-cddz6)     ;   **<< Do not round! >>** ;

   avgz1 =   S4671avg * &wtz1;
   avgz2 = ((S1048avg + S6635avg)/2) * &wtz2;
   avgz3 = ((S5114avg + S6175avg + S7905avg)/3) * &wtz3;
   avgz4 = ((S0442avg + S6624avg + S9367avg)/3) * &wtz4;
   avgz5 = ((S1194avg + S6719avg + S7050avg + S7306avg)/4) * &wtz5;
   avgz6 = ((S7888avg + S5115avg)/2) * &wtz6;
  **avg   = round(sum (of avgz1-avgz6),0.1);
   avg   =      sum (of avgz1-avgz6)     ;  **<< Do not round! >>** ;

 run ;

*;
```

```
/***********

libname out2 'S:\Weather\2016Tcap-Phase II\SoCalGas\';


**** Prelim and Final (Daily Data!) HDD, CDD and AVG together. ****;

data out2.DlySys_D ;
     set finalD prelimD ;  **<< Save a copy of DAILY Hdd, Cdd and Avg data for later use. >>** ;

  if (date < mdy(01,01,2015)) ;  **<< Complete DAILY data NOW through month of December 2014! >>**;

  keep date hdd cdd avg ;
run;


proc contents data=out2.DlySys_D ;
run ;




**** Prelim and Final HDD, CDD and AVG (Daily Data!) together--BY Station! ****;

data out2.DlyStn_D ;  **<< Save a copy of the Hdd by Station for later use. >>** ;
   set finalD prelimD ;
  if (date < mdy(01,01,2015)) ;  **<< Complete DAILY data NOW through month of December 2014! >>**;
run ;


proc contents data=out2.DlyStn_D ;
run ;
***********/






/***********
***********/

* Commented Out by GDT;
*libname out3 'C:\Weather\2016Tcap-Phase II\SoCalGas\';   **<< Alternative directory on my PERSONAL
Share-Drive! >>** ;

**** Prelim and Final HDD, CDD and AVG (Daily Data!) together--BY Station! ****;

data scgwea.DlySys_D ;
     set finalD prelimD ;  **<< Save a copy of DAILY Hdd, Cdd and Avg data for later use. >>** ;
```

```
  if (date < mdy(01,01,2022)) ;  **<< Complete DAILY data NOW through month of December 2021! >>**;

  keep date hdd cdd avg ;
run;


proc contents data=scgwea.DlySys_D ;
run ;



data scgwea.DlyStn_D ;  **<< Save a copy of the Hdd by Station for later use. >>** ;
   set finalD prelimD ;
  if (date < mdy(01,01,2022)) ;  **<< Complete DAILY data NOW through month of December 2021! >>**;
run ;


proc contents data=scgwea.DlyStn_D ;
run ;




quit ;
```

SoCalGas SAS Code #2: SysAvgVar(Monthly).sas

```
Title1 "Create MONTHLY Summaries of System Average HDD and CDD data from previousl created
DAILY" ;
Title2 "system average variables.  Calc. these summaries for System-Wide and by 15 Component Stations."
;

/*********************************************************************************/
/*                                                                               */
/*                                                                               */
/* Current Version of program file for our 2022 CGR work is:                     */
/*                                                                               */
/* Saved at: "/EDS_RB/Weather/2022 CGR/SoCalGas/SysAvgVar(Monthly).sas"  */
/*                                                                               */
/*              on Mar 13, 2022.                                                 */
/*                                                                               */
/*                                                                               */
/* Origial AUTHOR: Loan Nguyen, on September 12, 2001                            */
/*    from file: "s:\Weather\2003Bcaphdd(LXN).sas"                               */
/*                                                                               */
/*********************************************************************************/




options mprint ;
/* %cour8p
   %cour8l */


 options ls=211 ps=69 ; **<<LANDSCAPE:  SAS-Monospace w/Roman 6pt. Font >>**;
*options ls=160 ps=90 ; **<<PORTRAIT:   SAS-Monospace w/Roman 6pt. Font >>**;

options date number notes ;


**** Calculate Monthly Summaries for Hdd, Cdd from daily data sets! ****;

libname scgdf '/EDS_RB/Weather/2022 CGR/SoCalGas';

libname scgwea '/EDS_RB/Weather/2022 CGR/SoCalGas';


proc contents data=scgwea.DlyStn_D ;
run ;


%global wtz1 wtz2 wtz3 wtz4 wtz5 wtz6 ;



**<< Zone Weights a/o December 2021.      >>** ;
**<< (Data kindly provided by Andrew Tung. >>** ;
%let wtz1=0.005730 ; **<UPDATE>;
%let wtz2=0.038600 ; **<UPDATE>;
%let wtz3=0.182121 ; **<UPDATE>;
%let wtz4=0.072204 ; **<UPDATE>;
%let wtz5=0.381874 ; **<UPDATE>;
%let wtz6=0.319470 ; **<UPDATE>;
```

```
/*********************************************

**<< Zone Weights a/o December 2019.      >>** ;
**<< (Data kindly provided by Andrew Tung. >>** ;
%let wtz1=0.005632 ; **<UPDATE>;
%let wtz2=0.037582 ; **<UPDATE>;
%let wtz3=0.179423 ; **<UPDATE>;
%let wtz4=0.070095 ; **<UPDATE>;
%let wtz5=0.373126 ; **<UPDATE>;
%let wtz6=0.334142 ; **<UPDATE>;


**<< Zone Weights a/o December 2017.      >>** ;
**<< (Data kindly provided by Andrew Tung/Idan Enright. >>** ;
%let wtz1=0.005822 ;
%let wtz2=0.038497 ;
%let wtz3=0.185384 ;
%let wtz4=0.071615 ;
%let wtz5=0.383107 ;
%let wtz6=0.315576 ;


**<< Zone Weights a/o December 2015.      >>** ;
**<< (Data kindly provided by Idan Enright. >>** ;
%let wtz1=0.005842 ;
%let wtz2=0.038341 ;
%let wtz3=0.186160 ;
%let wtz4=0.071125 ;
%let wtz5=0.385713 ;
%let wtz6=0.312819 ;


**<< Zone Weights a/o December 2014.      >>** ;
**<< (Data kindly provided by Idan Enright. >>** ;
%let wtz1=0.0061 ;
%let wtz2=0.0423 ;
%let wtz3=0.1763 ;
%let wtz4=0.0747 ;
%let wtz5=0.3812 ;  **<< wtz5 calcd value was 0.3813, but used 0.3812 so total adds to 100.00 exactly.
>>** ;
%let wtz6=0.3194 ;

**<< Zone Weights a/o December 2013.      >>** ;
**<< (Data kindly provided by Andrew Tung. >>** ;
%let wtz1=0.0062 ;
%let wtz2=0.0424 ;
%let wtz3=0.1772 ;
%let wtz4=0.0742 ;
%let wtz5=0.3792 ;  **<< wtz5 calcd value was 0.3793, but used 0.3792 so total adds to 100.00 exactly.
>>** ;
%let wtz6=0.3208 ;

**<< New Weights for 6-Climate Zones: (per Avg. Mtr. Cnt for Yr)        >>** ;
**<< Zone Weights a/o December 2010.      >>** ;
**<< (Data kindly provided by Andrew Tung. >>** ;
%let wtz1=0.0062 ;
%let wtz2=0.0418 ;
%let wtz3=0.1774 ;
%let wtz4=0.0746 ;
%let wtz5=0.3802 ;
```

```
%let wtz6=0.3198 ;  **<< wtz6 calcd value was 0.3199, but used 0.3198 so total adds to 100.00 exactly.
>>** ;

**<< Zone Weights a/o December 2009.      >>** ;
**<< (Data kindly provided by Steve Tung. >>** ;
%let wtz1=0.0062 ;
%let wtz2=0.0417 ;
%let wtz3=0.1779 ;
%let wtz4=0.0745 ;
%let wtz5=0.3794 ;
%let wtz6=0.3203 ;

**<< Zone Weights a/o Year-2006.          >>** ;
**<< (Data kindly provided by Steve Tung. >>** ;
%let wtz1=0.0057 ;
%let wtz2=0.0354 ;
%let wtz3=0.1888 ;
%let wtz4=0.0676 ;
%let wtz5=0.3854 ;
%let wtz6=0.3171 ;

**<< Zone Weights a/o Year-2002.          >>** ;
**<< (Data kindly provided by Steve Tung. >>** ;
%let wtz1=0.0062 ;
%let wtz2=0.0332 ;
%let wtz3=0.1998 ;
%let wtz4=0.0662 ;
%let wtz5=0.3807 ;
%let wtz6=0.3139 ;

**<< Zone Weights a/o Year-2001.          >>** ;
**<< (Data kindly provided by Steve Tung. >>** ;
%let wtz1=0.0062 ;
%let wtz2=0.0320 ;
%let wtz3=0.2021 ;
%let wtz4=0.0638 ;
%let wtz5=0.3809 ;
%let wtz6=0.3150 ;

**<< Zone Weights a/o Year-2000.          >>** ;
**<< (Data kindly provided by Steve Tung. >>** ;
%let wtz1=0.0063 ;
%let wtz2=0.0312 ;
%let wtz3=0.2020 ;
%let wtz4=0.0655 ;
%let wtz5=0.3799 ;
%let wtz6=0.3151 ;

**<< Zone Weights a/o Year-1992. >>** ;
%let wtz1=0.0065 ;
%let wtz2=0.0282 ;
%let wtz3=0.1900 ;
%let wtz4=0.0620 ;
%let wtz5=0.3862 ;
%let wtz6=0.3271 ;
*********************************************/



**<< Code inserted to do the daily "system avg." calc. from Six-Zone formulas  >>**;
```

```
**<< that are used to calc. the daily "system avg." values for the NWS data.   >>**;

data DlyData ;
   set scgwea.DlyStn_D ;



***<< Calc. Climate Zones and "System Average" values for:         >>**;
***<<         Hdd, Cdd and Avg (temperature)                       >>**;
**  hddz1 =   S4671hdd * &wtz1;
**  hddz2 = ((S1048hdd + S6635hdd)/2) * &wtz2;
**  hddz3 = ((S5114hdd + S6175hdd + S7905hdd)/3) * &wtz3;
**  hddz4 = ((S0442hdd + S6624hdd + S9367hdd)/3) * &wtz4;
**  hddz5 = ((S1194hdd + S6719hdd + S7050hdd + S7306hdd)/4) * &wtz5;
**  hddz6 = ((S7888hdd + S5115hdd)/2) * &wtz6;
****hdd   = round(sum (of hddz1-hddz6),0.1);
**  hdd   =       sum (of hddz1-hddz6)     ;  **<< Do not round! >>** ;
** ;
**  cddz1 =   S4671cdd * &wtz1;
**  cddz2 = ((S1048cdd + S6635cdd)/2) * &wtz2;
**  cddz3 = ((S5114cdd + S6175cdd + S7905cdd)/3) * &wtz3;
**  cddz4 = ((S0442cdd + S6624cdd + S9367cdd)/3) * &wtz4;
**  cddz5 = ((S1194cdd + S6719cdd + S7050cdd + S7306cdd)/4) * &wtz5;
**  cddz6 = ((S7888cdd + S5115cdd)/2) * &wtz6;
****cdd   = round(sum (of cddz1-cddz6),0.1);
**  cdd   =       sum (of cddz1-cddz6)     ;  **<< Do not round! >>** ;
** ;
**  avgz1 =   S4671avg * &wtz1;
**  avgz2 = ((S1048avg + S6635avg)/2) * &wtz2;
**  avgz3 = ((S5114avg + S6175avg + S7905avg)/3) * &wtz3;
**  avgz4 = ((S0442avg + S6624avg + S9367avg)/3) * &wtz4;
**  avgz5 = ((S1194avg + S6719avg + S7050avg + S7306avg)/4) * &wtz5;
**  avgz6 = ((S7888avg + S5115avg)/2) * &wtz6;
****avg   = round(sum (of avgz1-avgz6),0.1);
**  avg   =       sum (of avgz1-avgz6)     ;  **<< Do not round! >>** ;
** ;

**<< Recover Avg, Hdd and Cdd by TZone: >>** ;

  avgz0 = avg ;
  avgz1 = avgz1 / &wtz1 ;
  avgz2 = avgz2 / &wtz2 ;
  avgz3 = avgz3 / &wtz3 ;
  avgz4 = avgz4 / &wtz4 ;
  avgz5 = avgz5 / &wtz5 ;
  avgz6 = avgz6 / &wtz6 ;

  hddz0 = hdd ;
  hddz1 = hddz1 / &wtz1 ;
  hddz2 = hddz2 / &wtz2 ;
  hddz3 = hddz3 / &wtz3 ;
  hddz4 = hddz4 / &wtz4 ;
  hddz5 = hddz5 / &wtz5 ;
  hddz6 = hddz6 / &wtz6 ;

  cddz0 = cdd ;
  cddz1 = cddz1 / &wtz1 ;
  cddz2 = cddz2 / &wtz2 ;
  cddz3 = cddz3 / &wtz3 ;
  cddz4 = cddz4 / &wtz4 ;
  cddz5 = cddz5 / &wtz5 ;
  cddz6 = cddz6 / &wtz6 ;
```

```
    keep   year month date
          avg avgz0-avgz6
          hdd hddz0-hddz6
          cdd cddz0-cddz6

        S0442hdd S1048hdd S1194hdd S4671hdd
        S5085hdd S5114hdd S5115hdd S6175hdd
        S6624hdd S6635hdd S6719hdd S7050hdd
        S7306hdd S7851hdd S7888hdd S7905hdd S9367hdd

        S0442cdd S1048cdd S1194cdd S4671cdd
        S5085cdd S5114cdd S5115cdd S6175cdd
        S6624cdd S6635cdd S6719cdd S7050cdd
        S7306cdd S7851cdd S7888cdd S7905cdd S9367cdd   ;  **<< Drop ALL othe variables! >>** ;

run;


*;
proc means data=DlyData nway noprint;
    class year month;
    var hdd hddz0-hddz6
        S0442hdd S1048hdd S1194hdd S4671hdd
        S5085hdd S5114hdd S5115hdd S6175hdd
        S6624hdd S6635hdd S6719hdd S7050hdd
        S7306hdd S7851hdd S7888hdd S7905hdd S9367hdd
        cdd cddz0-cddz6
        S0442cdd S1048cdd S1194cdd S4671cdd
        S5085cdd S5114cdd S5115cdd S6175cdd
        S6624cdd S6635cdd S6719cdd S7050cdd
        S7306cdd S7851cdd S7888cdd S7905cdd S9367cdd;
    output out=smByStn  sum=;
run ;
*;


data smByZone ;
    set smByStn  ;

    hdd = round(hdd);
  ** hddz0 = hddz0 ;  **<< Do not round this variable! >>** ;
    hddz1 = round(hddz1);
    hddz2 = round(hddz2);
    hddz3 = round(hddz3);
    hddz4 = round(hddz4);
    hddz5 = round(hddz5);
    hddz6 = round(hddz6);

    cdd = round(cdd);
  ** cddz0 = cddz0 ;  **<< Do not round this variable! >>** ;
    cddz1 = round(cddz1);
    cddz2 = round(cddz2);
    cddz3 = round(cddz3);
    cddz4 = round(cddz4);
    cddz5 = round(cddz5);
    cddz6 = round(cddz6);

    keep year month hdd cdd hddz0-hddz6  cddz0-cddz6 ;
run;
```

```
data ByStn ;
   set smByStn ;

  dateYYMM = mdy(month,1,year) ;

  if (dateYYMM < mdy(1,1,2022)) ;  **<< Only Pre Jan-2022 observations! >>** ;

  hdd = round(hdd) ;
  cdd = round(cdd) ;

 **<< The "labeling" below is based on the pre-1997 data from NWS! >>** ;
  label s5115hdd = "LA Civic Cntr." ;
  label s5114hdd = "LA Air Port" ;
  label s1194hdd = "Burbank" ;
  label s7050hdd = "Pomona" ;
  label s6719hdd = "Pasadena" ;
  label s7306hdd = "Redlands" ;
  label s6175hdd = "Newport Beach" ;
  label s7888hdd = "Santa Ana" ;
  label s4671hdd = "Lake Arrowhead" ;
  label s6624hdd = "Palmdale" ;
  label s1048hdd = "Brawley" ;
  label s6635hdd = "Palmlabel springs" ;
  label s7905hdd = "Santa Barbara-Apt" ;
  label s0442hdd = "Bakersfield" ;
  label s9367hdd = "Visalia" ;

  label s5085hdd = "Long Beach(Apt)" ;
  label s7851hdd = "San Luis Obispo(@CSU)" ;

  label hdd = "Syst-Avg. Hdd" ;
run ;


proc sort data=ByStn ;
   by year month ;
run ;


proc print data=ByStn  uniform split="/" ;
 id year;
 by year;
 sumby year;
    var dateYYMM hdd
        S4671hdd
        S1048hdd   S6635hdd
        S5114hdd   S6175hdd   S7905hdd
        S0442hdd   S6624hdd   S9367hdd
        S1194hdd   S6719hdd   S7050hdd   S7306hdd
        S7888hdd   S5115hdd  ;
    **  s5085hdd   s7851hdd   ;
    sum hdd
        S4671hdd
        S1048hdd   S6635hdd
        S5114hdd   S6175hdd   S7905hdd
        S0442hdd   S6624hdd   S9367hdd
```

```
        S1194hdd   S6719hdd   S7050hdd   S7306hdd
        S7888hdd   S5115hdd  ;
    **  s5085hdd   s7851hdd   ;
    format  dateYYMM worddate3.  hdd
        S4671hdd
        S1048hdd   S6635hdd
        S5114hdd   S6175hdd   S7905hdd
        S0442hdd   S6624hdd   S9367hdd
        S1194hdd   S6719hdd   S7050hdd   S7306hdd
        S7888hdd   S5115hdd   6. ;
    **  s5085hdd   s7851hdd   6. ;

  label s5115hdd = "Los Angeles/ Civic Cntr/Zone6/--------"
       s5114hdd = "Los Angeles/ (Apt) /Zone3/--------"
       s1194hdd = "Burbank /Zone5/--------"
       s7050hdd = "Pomona /Zone5/--------"
       s6719hdd = "Pasadena /Zone5/--------"
       s7306hdd = "Redlands /Zone5/--------"
       s6175hdd = "Newport /Beach /Zone3/--------"
       s7888hdd = "Santa Ana /Zone6/--------"
       s4671hdd = "Lake Arrow-/ head /Zone1/--------"
       s6624hdd = "Palmdale /Zone4/--------"
       s1048hdd = "Brawley /Zone2/--------"
       s6635hdd = "Palm Springs /Zone2/--------"
       s7905hdd = "Santa Bar-/bara(Apt) /Zone3/--------"
       s0442hdd = "Bakers-/ field /Zone4/--------"
       s9367hdd = "Visalia /Zone4/--------"

       hdd = "Syst-Avg./ Hdd /Zone1-6/--------" ;

 **<< Note: The last two("s5085hdd" and "s7851hdd") are not in the "Sys-Avg" calc. >>** ;
 **<<       The first six(6) sets are used in Climate Zones 1-6, respectively.     >>** ;
 **<<       The variable "hdd" is the System-Average value from Climate Zones 1-6. >>** ;
 **  s5085hdd = "Long Beach/ (Apt) /No-Zone/--------"        ;
 **  s7851hdd = "San Luis /Obispo(@CSU) /No-Zone/--------"   ;

run ;



proc means data=ByStn noprint ;
   class year ;
    var hdd
        S4671hdd
        S1048hdd   S6635hdd
        S5114hdd   S6175hdd   S7905hdd
        S0442hdd   S6624hdd   S9367hdd
        S1194hdd   S6719hdd   S7050hdd   S7306hdd
        S7888hdd   S5115hdd  ;
     ** s5085hdd   s7851hdd   ;
    output out=SumByYr sum=;
run ;

/**
data SumByYr ;
  set SumByYr ;
  if (year=.) then delete ;
run ;
**/

proc print data=SumByYr  uniform split="/" ;
  where year ne . ;
```

```
    var year hdd
       S4671hdd
       S1048hdd   S6635hdd
       S5114hdd   S6175hdd   S7905hdd
       S0442hdd   S6624hdd   S9367hdd
       S1194hdd   S6719hdd   S7050hdd   S7306hdd
       S7888hdd   S5115hdd  ;
     ** s5085hdd   s7851hdd   ;
    format   hdd
       S4671hdd
       S1048hdd   S6635hdd
       S5114hdd   S6175hdd   S7905hdd
       S0442hdd   S6624hdd   S9367hdd
       S1194hdd   S6719hdd   S7050hdd   S7306hdd
       S7888hdd   S5115hdd   6. ;
     ** s5085hdd   s7851hdd   6. ;

  label s5115hdd = "Los Angeles/ Civic Cntr/Zone6/--------"
       s5114hdd = "Los Angeles/ (Apt) /Zone3/--------"
       s1194hdd = "Burbank /Zone5/--------"
       s7050hdd = "Pomona /Zone5/--------"
       s6719hdd = "Pasadena /Zone5/--------"
       s7306hdd = "Redlands /Zone5/--------"
       s6175hdd = "Newport /Beach /Zone3/--------"
       s7888hdd = "Santa Ana /Zone6/--------"
       s4671hdd = "Lake Arrow-/ head /Zone1/--------"
       s6624hdd = "Palmdale /Zone4/--------"
       s1048hdd = "Brawley /Zone2/--------"
       s6635hdd = "Palm Springs /Zone2/--------"
       s7905hdd = "Santa Bar-/bara(Apt) /Zone3/--------"
       s0442hdd = "Bakers-/ field /Zone4/--------"
       s9367hdd = "Visalia /Zone4/--------"

       hdd = "Syst-Avg./ Hdd /Zone1-6/--------" ;

 **<< Note: The last two("s5085hdd" and "s7851hdd") are not in the "Sys-Avg" calc. >>** ;
 **<<      The first six(6) sets are used in Climate Zones 1-6, respectively.     >>** ;
 **<<      The variable "hdd" is the System-Average value from Climate Zones 1-6. >>** ;
 **  s5085hdd = "Long Beach/ (Apt) /No-Zone/--------"        ;
 **  s7851hdd = "San Luis /Obispo(@CSU) /No-Zone/--------"   ;

run ;
```

```
**** Prelim and final HDD and CDD together ****;
data all;
    set smByZone ;
  date = mdy(month,01,year) ;  format date monyy7. ;
  if (date < mdy(01,01,2022)) ;  **<< Complete Monthly data only through month of December 2021. >>**;
run;

proc sort data=all out=monthly;
    by year month;
run;
```

```
/***********
***********/
data scgwea.mn50_21 ;  **<< Save a copy of the Monthly Hdd/Cdd data for later use. >>** ;
   set monthly ;
 **  keep date year month hdd cdd ;   **<< Keep all zone data as well as system average data. >>** ;
 **<< Export a copy as ".xls" as well. >>** ;
run ;

data scgwea.ByStn ;  **<< Save a copy of the Hdd by Station for later use. >>** ;
   set ByStn ;
 **<< Export a copy as ".xls" as well. >>** ;
run ;




***<< Print Summary Tables of Hdd and Cdd by month  with Annual Totals >>*** ;

proc transpose data=monthly out=hddSmry prefix=mon ;
   by year;
   id month ;
   var hdd ;
run ;

data hddSmry ;
   set hddSmry ;
 hddTot = sum(of mon1-mon12) ;
run ;

proc print data=hddSmry ;
 id year ;
 var hddTot mon1-mon12 ;
title1 'Monthly Heating Degree-Days from 1950 thru 2021 (Month-to-Date)';
title2 "   " ;
run ;



proc transpose data=monthly out=cddSmry prefix=mon ;
   by year;
   id month ;
   var cdd ;
run ;

data cddSmry ;
   set cddSmry ;
 cddTot = sum(of mon1-mon12) ;
run ;

proc print data=cddSmry ;
 id year ;
 var cddTot mon1-mon12 ;
title1 'Monthly Cooling Degree-Days from 1950 thru 2021 (Month-to-Date)';
run ;
```

```
/********  BEGIN:  Surpress execution of Stat. Analysis **************************
proc print data=monthly;
     where 1950 <= year <= 2014;
     id year month;
     var date hdd cdd ;
title1 'Monthly Heating/Cooling degree days from 1950 thru 2014(Month-to-Date)';
run;


proc plot data=monthly ;
     where 1950 <= year <= 1959;
    plot hdd*date='*' ;
    plot cdd*date='+' ;
title1 'Plot of MONTHLY Heating/Cooling degree days from 1950 thru 1959';
run ;

proc plot data=monthly ;
     where 1960 <= year <= 1969;
    plot hdd*date='*' ;
    plot cdd*date='+' ;
title1 'Plot of MONTHLY Heating/Cooling degree days from 1960 thru 1969';
run ;

proc plot data=monthly ;
     where 1970 <= year <= 1979;
    plot hdd*date='*' ;
    plot cdd*date='+' ;
title1 'Plot of MONTHLY Heating/Cooling degree days from 1970 thru 1979';
run ;

proc plot data=monthly ;
     where 1980 <= year <= 1989;
    plot hdd*date='*' ;
    plot cdd*date='+' ;
title1 'Plot of MONTHLY Heating/Cooling degree days from 1980 thru 1989';
run ;

proc plot data=monthly ;
     where 1990 <= year <= 1999;
    plot hdd*date='*' ;
    plot cdd*date='+' ;
title1 'Plot of MONTHLY Heating/Cooling degree days from 1990 thru 1999';
run ;

proc plot data=monthly ;
     where 2000 <= year <= 2009;
    plot hdd*date='*' ;
    plot cdd*date='+' ;
title1 'Plot of MONTHLY Heating/Cooling degree days from 2000 thru 2009';
run ;

proc plot data=monthly ;
     where 2010 <= year <= 2014;
    plot hdd*date='*' ;
    plot cdd*date='+' ;
title1 'Plot of MONTHLY Heating/Cooling degree days from 2010 thru 2014';
run ;
```

```
proc means data=all nway ;   ** noprint;
    class year;
    var hdd cdd ;
    output out=year sum=hdd cdd ;
title1 'Summary Statistics by YEAR:' ;
run;

proc print data=year;
    where 1950 <= year <= 2014;
    id year;
    var hdd cdd;
    sum hdd cdd;
title1 'Annual Heating/Cooling degree days from 1950 thru 2014';

proc plot data=year ;
    where 1950 <= year <= 2014;
   plot hdd*year='*' ;
   plot cdd*year='+' ;
title1 'Plot of ANNUAL Heating/Cooling degree days from 1950 thru 2014';
run ;


run;

proc means data=monthly nway ;   ** noprint;
    where 1950 <= year <= 2014;
    class month;
    var hdd cdd;
    output out=avg mean=;
title1 'Summary Statistics by MONTH (for years 1950-2014):' ;
run;

proc print;
    id month;
    var hdd cdd ;
    sum hdd cdd ;
title1 '65-Year Average Monthly Heating Degree Days';
run;

**** "65" <<year standard deviation>> ****;
proc means data=year ;   **noprint;
    where 1950 <= year <= 2014;
    var hdd cdd;
    output out=std65 std=Hstd65 Cstd65;
run;

**** "65" <<year "normal" as Average Year>> ****;
proc means data=year ;   **noprint;
    where 1950 <= year <= 2014;
    var hdd cdd ;
    output out=avg65 mean=hdd cdd ;
run;

**** Small sample - Use t-statistics with 65-1 DF ****;
**** 1 in 35 probability for "HOT" and "COLD" years ****;
data finalyr;
    set avg65;
    if _n_ = 1 then set std65;
    p35   = 1-(1/35);
    tstat = tinv(p35,65-1);
*    zstat = probit(p35);
```

```
    Havgyr = round(HDD);
    Hcolyr = round(Havgyr + (tstat * Hstd65)); **<< "More" Hdd means LOWER temps.! >>**;
    Hhotyr = round(Havgyr - (tstat * Hstd65);

    Cavgyr = round(CDD);
    Ccolyr = round(Cavgyr - (tstat * Cstd65));
    Chotyr = round(Cavgyr + (tstat * Cstd65)); **<< "More" Cdd means HIGHER temps.! >>**;
run;

data finalmo;
    set avg(rename=(hdd=Havg_mo cdd=Cavg_mo));
    if _n_ = 1 then set finalyr;

    Hratio  = Havg_mo / Hdd;
    Havg_mo = Hratio * Havgyr;
    Hhot_mo = Hratio * Hhotyr;
    Hcol_mo = Hratio * Hcolyr;

    Cratio  = Cavg_mo / Cdd;
    Cavg_mo = Cratio * Cavgyr;
    Chot_mo = Cratio * Chotyr;
    Ccol_mo = Cratio * Ccolyr;

    drop _type_ _freq_ p35;
run;

proc print;
    id month;
    var Havgyr Hcolyr Hhotyr Havg_mo Hcol_mo Hhot_mo tstat Hstd65;
    format Havg_mo Hcol_mo Hhot_mo 9.4;
    sum Havg_mo Hhot_mo Hcol_mo;
title1 'Avg, Cold and Hot Year Heating Degree-Days ';
run;


proc print;
    id month;
    var Cavgyr Ccolyr Chotyr Cavg_mo Ccol_mo Chot_mo tstat Cstd65;
    format Cavg_mo Ccol_mo Chot_mo 9.4;
    sum Cavg_mo Chot_mo Ccol_mo;
title1 'Avg, Cold and Hot Year Cooling Degree-Days ';
run;

*********  END:    Surpress execution of Stat. Analysis *************************/






quit ;
```

SoCalGas SAS Code #3: MinAvg-Freq(SoCalGas)ByMonth.sas

```
title1 "Calculate Min{Avg} (Minimum Average Daily Temp.) by Months for all data over a specified range of
YEARS." ;

/**********************************************************************************

   File: S:\Weather\2022 CGR\SoCalGas\MinAvg-Freq(SoCalGas)ByMonth.sas

 **********************************************************************************/


options date number source notes ;
options mprint ;

/* %cour8l
   %cour8p; */

 options ls=160 ps=90 ;  *<To get PORTRAIT  and SAS-Monospace w/Roman 6pt. FONT >*;
*options ls=211 ps=69 ;  *<To get LANDSCAPE and SAS-Monospace w/Roman 6pt. FONT >*;

**options nomprint ;


     **<< Data set "DlySys_d.sas7bdat" was created by SAS program on file:                >>** ;
     **<<      "S:\Weather\2022 CGR\SoCalGas\SysAvgVar(Daily).sas"                                 >>** ;
     **<<                                                        >>** ;
libname in '/EDS_RB/Weather/2022 CGR/SoCalGas';



   **<< Directory to save a copy of output data set! >>** ;
libname out '/EDS_RB/Weather/2022 CGR/SoCalGas';


proc contents data=in.DlySys_d ;
run ;



%let startyr=1950;  ***<< Value of "Start Year" >>***;
%let lastyr=2021 ;  ***<< Value of "Last Year" >>***;

/********
%let tgtmonth=xx ;  ***<< Value of "Target Month", i.e., 1,2,3, ... 12 >>** ;
********/




proc format ;
   value mmmmmm    1='Jan'
                2='Feb'
                3='Mar'
                4='Apr'
                5='May'
                6='Jun'
                7='Jul'
                8='Aug'
                9='Sep'
               10='Oct'
               11='Nov'
```

```
                12='Dec'
                13='Min4Yr' ;
run ;


%macro FreqMon(name_mon,tgtmonth) ;

data combined ;
    set in.DlySys_d ;

year = year(date) ;
month= month(date) ;
day = day(date) ;
  **<< To "Select" Winter Months Only. >>**;
  **  if month in (1,2,3,11,12);

*** hdd = round(avg,1);  **<< Comment out ... Do NOT round data this time! >>** ;
if ((year >= &startyr) & (year <= &lastyr)) ;

%if (&tgtmonth >= 1 and &tgtmonth <= 12) %then
  %do ;
if (month = &tgtmonth) ;  **<< To select only a specific month! >>** ;
  %end ;
%else
  %do ;
month = 13 ;  **<< Set "month" variable to "13" and select ALL "months" of the YEAR! >>** ;
  %end ;
run;


proc sort data=combined;
 by year month day ;
run;

***proc contents data=combined ;
***run ;

proc means data=combined noprint;
  by year month ;
  var avg ;
  output out=&name_mon min=MinAvg;
  title1 "Minium{avg} (Minimum Avg. Daily Temp.) for &name_mon  by YEAR=&startyr to &lastyr";
run;

proc print data=&name_mon ;
run ;

%mend ;








%let startyr=1950;  ***<< Value of "Start Year" >>***;
%let lastyr=2021 ;  ***<< Value of "Last Year" >>***;
```

```
/************
%let name_mon='January ';  ***<< NAME of "Target Month" >>***;
%let tgtmonth= 1 ;    ***<< Value of "Target Month" >>***;
************/

%FreqMon(JAN,1) ;

data AllMonth ;
  set JAN ;
run ;


%FreqMon(FEB,2) ;

data AllMonth ;
  set AllMonth FEB ;
run ;


%FreqMon(MAR,3) ;

data AllMonth ;
  set AllMonth MAR ;
run ;


%FreqMon(APR,4) ;

data AllMonth ;
  set AllMonth APR ;
run ;


%FreqMon(MAY,5) ;

data AllMonth ;
  set AllMonth MAY ;
run ;


%FreqMon(JUN,6) ;

data AllMonth ;
  set AllMonth JUN ;
run ;


%FreqMon(JUL,7) ;

data AllMonth ;
  set AllMonth JUL ;
run ;


%FreqMon(AUG,8) ;

data AllMonth ;
  set AllMonth AUG ;
run ;


%FreqMon(SEP,9) ;
```

```
data AllMonth ;
  set AllMonth SEP ;
run ;


%FreqMon(OCT,10) ;

data AllMonth ;
  set AllMonth OCT ;
run ;


%FreqMon(NOV,11) ;

data AllMonth ;
  set AllMonth NOV ;
run ;


%FreqMon(DEC,12) ;

data AllMonth ;
  set AllMonth DEC ;
run ;


%FreqMon(ALL,13) ;

data AllMonth ;
  set AllMonth All ;
run ;



proc Tabulate data=AllMonth;
  class year month ;
  var MinAvg ;
  table year, MinAvg*f=6.2*(month)/rts=6 ;
  label MinAvg='Min{Avg} by Mo' ;
 *keylabel All ='Min4Yr';
  format month mmmmmm. ;
  title2 "Min{avg} by Months for all Years from YEAR=&startyr to &lastyr";
run;


proc sort data=AllMonth out=MinAvg_d;
  by month year MinAvg ;
run ;


proc print data=MinAvg_d ;
  by month ;
  pageby month ;
  var year MinAvg ;
run ;


data out.MinAvg_d ;  *<< Save a copy for later use!  >>** ;
  set MinAvg_d ;
 drop _freq_ _type_ ;
run ;
```

```
proc contents data=out.MinAvg_d ;
run ;


/************************
 **<< Export a copy as ".dbf" as well. >>** ;
 **<< Note: Must "delete" the prior version, othewise the save will not execute. >>** ;
proc dbload dbms=dbase data=MinAvg_d ;
path='S:\Weather\2016Tcap-Phase II\SoCalGas\MinAvg_d.dbf';
limit=0;
load;
run;
*************************/



quit ;
```

SoCalGas SAS Code #4: GEV4DlyTemp(NLReg2)ByMonthMACRO_Scg.sas

```
Title1 "Data Analysis for Maximum/Minimum Daily SysAvg Temperatures (Un-Rounded)." ;
Title2 "Fit GEV, Double-Exp and T-Dist Probability Models to Empirical CDF using NL-OLS Regression
Methods." ;

/*****************************************************************************************************/
/*                                                                                         */
/*                                                                                         */
/*                                                                                         */
/* FILE SAVED: "S:\Weather\2022 CGR\SoCalGas\GEV4DlyTemp(NLReg2)ByMonthMACRO_Scg.sas"
                                              */
/*          Purpose: Annual Max of Negative of Min. Temp.                                          */
/*          for each of calendar months Jan-Dec, and the entire year(index=13).            */
/*                                                                                         */
/*          Fit GEV models (3-parameter and 2-parameter), plus a simple T-Distribution model.       */
/*                                                                                         */
/*                                                                                         */
/*****************************************************************************************************/




options mprint ;
/* %cour8p
   %cour8l */


 options ls=211 ps=69 ; **<<LANDSCAPE:  SAS-Monospace w/Roman 6pt. Font >>**;
*options ls=160 ps=90 ; **<<PORTRAIT:   SAS-Monospace w/Roman 6pt. Font >>**;

options date number notes ;



libname out1 '/EDS_RB/Weather/2022 CGR/SoCalGas' ;  **<< Directory for daily weather variables as
INPUT. >>**;

libname out2 '/EDS_RB/Weather/2022 CGR/SoCalGas/MinTemp' ;  **<< Directory for estimation results
OUPUT files. >>** ;


proc contents data=out1.DlySys_d ;
run ;

data seriesD ;
  set out1.DlySys_d ;
  year = year(date) ;
  month = month(date) ;
  posAvg = avg ;
  negAvg = -avg ;
run ;


proc means data=seriesD noprint nway ;
 class year month ;
  var posAvg negAvg ;
 output out=mostat
                mean=posAvg  negAvg
```

```
                max=MxPosAvg MxNegAvg
                min=MnPosAvg MnNegAvg    ;
run;


proc sort data=mostat ;
  by year month ;
run ;


data mostat ;
  set mostat ;
  MxPRatio = MxPosAvg/ PosAvg  ;
  MnPRatio = MnPosAvg/ PosAvg  ;
  MxNRatio = MxNegAvg/ NegAvg  ;
  MnNRatio = MnNegAvg/ NegAvg  ;
run ;






/***************
***<< Print Summary Tables of Means/Minimums/Maximums of daily NEGATIVE-Temperatures (degrees-F).
>>*** ;

proc transpose data=mostat out=AvTData prefix=AvT_ ;
  where (year < 2015) ;
  by year;
  id month ;
  var NegAvg  ;
run ;

data AvTData ;
  set AvTData ;

 if (mod(year,4)=0) then do ;
       AvT_13 = (AvT_1 + AvT_3 + AvT_5 + AvT_7 + AvT_8 + AvT_10 + AvT_12)*31
             + (AvT_4 + AvT_6 + AvT_9 + AvT_11)*30
             + (AvT_2)*29 ;
       AvT_13 = AvT_13 / 366 ;
       end ;
 else do ;
       AvT_13 = (AvT_1 + AvT_3 + AvT_5 + AvT_7 + AvT_8 + AvT_10 + AvT_12)*31
             + (AvT_4 + AvT_6 + AvT_9 + AvT_11)*30
             + (AvT_2)*28 ;
       AvT_13 = AvT_13 / 365 ;
       end ;

run ;

proc print data=AvTData ;
 id year ;
 var AvT_13 AvT_1-AvT_12 ;
title3 'Monthly Mean NEGATIVE Temperature (Deg-F) from 1950 thru 2015.';
run ;
```

```
proc transpose data=mostat out=MnTData prefix=Mn2016T_ ;
   where (year < 2015) ;
   by year;
   id month ;
   var MnNegAvg ;
run ;

data MnTData ;
   set MnTData ;
 MnT_13 = min(of MnT_1-MnT_12) ;
run ;

proc print data=MnTData ;
 id year ;
 var MnT_13 MnT_1-MnT_12 ;
title3 'Monthly MINIMUM NEGATIVE-Temperature (Deg-F) from 1950 thru 2015.';
run ;
****************/




proc transpose data=mostat out=MxTData prefix=MxT_ ;
   where (year < 2022) ;
   by year;
   id month ;
   var MxNegAvg ;
run ;

data MxTData ;
   set MxTData ;
 MxT_13 = max(of MxT_1-MxT_12) ;
run ;

proc print data=MxTData ;
 id year ;
 var MxT_13 MxT_1-MxT_12 ;
title3 'Monthly MAXIMUM NEGATIVE-Temperature (Deg-F) from 1950 thru 2021.';
run ;




/****************
***<< Descriptive Statistics: Maximums of daily NEGATIVE-Temperatures (Deg-F) for Year and each
calendar month. >>*** ;


proc corr data=MxTData ;
  var MxTyr MxT_1 - MxT_12 ;
title3 'Correlation Matrix of Monthly Maximum NEGATIVE-Temperatures (Deg-F) within same year.';
run ;

proc arima data=MxTData ;
```

```
  identify var=MxT_13 ;
  identify var=MxT_1 ;
  identify var=MxT_2 ;
  identify var=MxT_3 ;
  identify var=MxT_4 ;
  identify var=MxT_5 ;
  identify var=MxT_6 ;
  identify var=MxT_7 ;
  identify var=MxT_8 ;
  identify var=MxT_9 ;
  identify var=MxT_10 ;
  identify var=MxT_11 ;
  identify var=MxT_12 ;
title3 "Auto-correlation analysis of each calendar month's Maximum NEGATIVE-Temperatures (Deg-F)
within same year.";
run ;

proc univariate normal data=MxTData plot ;
  id year ;
  var MxT_13 MxT_1 - MxT_12 ;
title3 "Probability plots and tests for NORMALity by each calendar month's Maximun NEGATIVE-
Temperatures (Deg-F) time series.";
run ;


proc means data=MxTData ;
  var MxT_1 - MxT_12  MxT_13 ;
run ;
****************/
```

```
***<< Statistical Estimation of GEV Models: Maximums of daily heating degrees for Year and each calendar
month. >>*** ;

%macro RankIt(file=MxTData,var=MxT_13,rank=Rank,prob=PrMxT_13,Nobser=72,PltValue=0.375) ;
proc sort data=&file ;
  by &var ;
run ;

data &file ;
   set &file ;
   retain &rank 0     alpha &pltvalue ;

  &rank = &rank + 1 ;
  &prob =  (&rank - alpha) / (&Nobser +(1 - 2*alpha)) ;
run ;

proc print data=&file ;
  var &var &rank &prob alpha year ;
run ;
%mend RankIt ;
```

```
%macro
GEVfit(file=MxTData,ofile=MxTNL1,outfit=fit1,outest=est1,depvar=PrMxT_13,var=MxT_13,typeGEV=1,
            KappaI=0.25,GammaI=-47.05,ThetaI=2.77,YrLo=1950,YrHi=2021) ;
proc sort data=&file ;
   by year ;
run ;



proc model data=&file converge=0.001
          maxit=500 dw ;  outmodel=&ofile ;
     range year = &YrLo to &YrHi ;  **<< Dropped any months of current year data. >>** ;


     y = (&var - Gamma) / Theta ;

    %if &typeGEV=1 %then %do ;  ***<< 3-parameter GEV Model. >>>*** ;
       &depvar = exp( -(1 - Kappa * (y))**(1/Kappa) ) ;
       %let typmod = 3-parameter GEV Model. ;
       %end ;

    %if &typeGEV=2 %then %do ;  **<< 2-parameter "Double Exponential" or  "Gumbel" Model. >>** ;
       &depvar = exp( -exp(-(y)) ) ;
       %let typmod = 2-parameter Double Exponential or Gumbel Model. ;
       %end ;

    %if (&typeGEV NE 1) AND (&typeGEV NE 2) %then  %do ; **<< 2-parameter "T-Dist" Model. >>** ;
       dft=(&YrHi - &YrLo) +1 -2 ;
       &depvar = probt(y,dft) ;
       %let typmod = 2-parameter T-Dist Model. ;
       %end ;


  %if &typeGEV = 1 %then %do ;
  parms
       Kappa  &KappaI
       Gamma  &GammaI
       Theta  &ThetaI ;
  %end ;

  %if (&typeGEV NE 1)  %then %do ;
  parms
       Gamma  &GammaI
       Theta  &ThetaI ;
  %end ;


  fit &depvar   /out=&outfit  outall
               outest=&outest corrb corrs outcov ;

title3 "Non-linear Estimation of &&typmod: for Maximum NEGATIVE Temperature (Deg-F)." ;
run ;
%mend GEVfit ;






%macro GEVbyMo(mm=_1) ;
```

```
*********<<< Analysis for "January"    (i.e., SUFIX "mm" = "_1"  >>>******************;
*********<<< Analysis for "February"   (i.e., SUFIX "mm" = "_2"  >>>******************;
*********<<< Analysis for "March"      (i.e., SUFIX "mm" = "_3"  >>>******************;
*********<<< Analysis for "April"      (i.e., SUFIX "mm" = "_4"  >>>******************;
*********<<< Analysis for "May"        (i.e., SUFIX "mm" = "_5"  >>>******************;
*********<<< Analysis for "June"       (i.e., SUFIX "mm" = "_6"  >>>******************;
*********<<< Analysis for "July"       (i.e., SUFIX "mm" = "_7"  >>>******************;
*********<<< Analysis for "August"     (i.e., SUFIX "mm" = "_8"  >>>******************;
*********<<< Analysis for "September"  (i.e., SUFIX "mm" = "_9"  >>>******************;
*********<<< Analysis for "October"    (i.e., SUFIX "mm" = "_10"  >>>******************;
*********<<< Analysis for "November"   (i.e., SUFIX "mm" = "_11"  >>>******************;
*********<<< Analysis for "December"   (i.e., SUFIX "mm" = "_12"  >>>******************;
*********<<< Analysis for "ALL Months"   (i.e., SUFIX "mm" = "_13"  >>>******************;

%RankIt(file=MxTData,var=MxT&mm,rank=Rank&mm,prob=PrMxT&mm,Nobser=72,PltValue=0.375) ;




%GEVfit(file=MxTData,ofile=MxTNL2,outfit=fit2,outest=est2,depvar=PrMxT&mm,var=MxT&mm,typeGEV=2,
        KappaI=0.25,GammaI=&&gamma&mm,ThetaI=&&theta&mm,YrLo=1950,YrHi=2021) ;


proc print data=fit2 ;
run ;




proc transpose data=fit2 out=pred2 prefix=probP_ ;
   where (_type_ = "PREDICT" ) ;
   by year;
   var prmxt&mm ;
run ;

data comb2 ;
   merge MxTData pred2 ;
  by year ;
   ProbP2 = ProbP_1 ;
  keep year MxT&mm PrMxT&mm ProbP2 ;
run ;


proc print data=comb2 ;
run ;


proc plot data=comb2 ;
  plot prmxt&mm*MxT&mm='*'
       probP2*MxT&mm='-' / overlay ;
run ;


proc print data=est2 ;
run ;



data out2.est2&mm ;   ***<<< Save a copy of the "Double Exponential Model" estimation results!  >>>*** ;
  set est2 ;
run ;
```

```
%GEVfit(file=MxTData,ofile=MxTNL0,outfit=fit0,outest=est0,depvar=PrMxT&mm,var=MxT&mm,typeGEV=0,
        KappaI=0.25,GammaI=&&gamma&mm,ThetaI=&&theta&mm,YrLo=1950,YrHi=2021) ;


proc print data=fit0 ;
run ;



proc transpose data=fit0 out=pred0 prefix=probP_ ;
   where (_type_ = "PREDICT" ) ;
   by year;
   var prmxt&mm ;
run ;

data comb0 ;
   merge MxTData pred0 ;
  by year ;
   ProbP0 = ProbP_1 ;
  keep year MxT&mm PrMxT&mm ProbP0 ;
run ;


proc print data=comb0 ;
run ;


proc plot data=comb0 ;
  plot prmxt&mm*MxT&mm='*'
       probP0*MxT&mm='-' / overlay ;
run ;


proc print data=est0 ;
run ;



data out2.est0&mm ;    ***<<< Save a copy of the 2-parameter "T-Distribution" Model estimation results!
>>>*** ;
  set est0 ;
run ;



%GEVfit(file=MxTData,ofile=MxTNL1,outfit=fit1,outest=est1,depvar=PrMxT&mm,var=MxT&mm,typeGEV=1,
        KappaI=0.25,GammaI=&&gamma&mm,ThetaI=&&theta&mm,YrLo=1950,YrHi=2021) ;


proc print data=fit1 ;
run ;


proc transpose data=fit1 out=pred1 prefix=probP_ ;
```

```
   where (_type_ = "PREDICT" ) ;
   by year;
   var prmxt&mm ;
run ;

data comb1 ;
   merge MxTData pred1 ;
  by year ;
   ProbP1 = ProbP_1 ;
  keep year MxT&mm PrMxT&mm ProbP1 ;
run ;


proc print data=comb1 ;
run ;


proc plot data=comb1 ;
  plot prmxt&mm*MxT&mm='*'
      ProbP1*MxT&mm='-' / overlay ;
run ;



proc print data=est1 ;
run ;



data out2.est1&mm ;   ***<<< Save a copy of the "G.E.V. Model" estimation results!  >>>*** ;
  set est1 ;
run ;

%mend GEVbyMo ;



/****************
****************/

proc means data=MxTData ;
  var MxT_1 - MxT_12  MxT_13;
 output out=VarStat
       mean=mean1-mean12 mean13
       std=stdev1-stdev12 stdev13;
title3 "Calc. Means and Standard Deviations to use as Starting Values in Non-Linear Estimations." ;
run ;


proc print data=VarStat ;
run ;

data _null_ ;
   set VarStat ;

  call symput('gamma_13',mean13) ;
  call symput('theta_13',stdev13) ;

  call symput('gamma_12',mean12) ;
  call symput('theta_12',stdev12) ;

  call symput('gamma_11',mean11) ;
```

```
  call symput('theta_11',stdev11) ;

  call symput('gamma_10',mean10) ;
  call symput('theta_10',stdev10) ;

  call symput('gamma_9',mean9) ;
  call symput('theta_9',stdev9) ;

  call symput('gamma_8',mean8) ;
  call symput('theta_8',stdev8) ;

  call symput('gamma_7',mean7) ;
  call symput('theta_7',stdev7) ;

  call symput('gamma_6',mean6) ;
  call symput('theta_6',stdev6) ;

  call symput('gamma_5',mean5) ;
  call symput('theta_5',stdev5) ;

  call symput('gamma_4',mean4) ;
  call symput('theta_4',stdev4) ;

  call symput('gamma_3',mean3) ;
  call symput('theta_3',stdev3) ;

  call symput('gamma_2',mean2) ;
  call symput('theta_2',stdev2) ;

  call symput('gamma_1',mean1) ;
  call symput('theta_1',stdev1) ;

run ;


%GEVbyMo(mm=_13) ;   **<< Annual(Entire Year of Data.) >>** ;


%GEVbyMo(mm=_1) ;   **<< Jan Data. >>** ;


%GEVbyMo(mm=_2) ;   **<< Feb Data. >>** ;
%GEVbyMo(mm=_3) ;   **<< Mar Data. >>** ;


%GEVbyMo(mm=_4) ;   **<< Apr Data. >>** ;


%GEVbyMo(mm=_5) ;   **<< May Data. >>** ;
%GEVbyMo(mm=_6) ;   **<< Jun Data. >>** ;
%GEVbyMo(mm=_7) ;   **<< Jul Data. >>** ;


%GEVbyMo(mm=_8) ;   **<< Aug Data. >>** ;


%GEVbyMo(mm=_9) ;   **<< Sep Data. >>** ;
%GEVbyMo(mm=_10) ;   **<< Oct Data. >>** ;
%GEVbyMo(mm=_11) ;   **<< Nov Data. >>** ;
%GEVbyMo(mm=_12) ;   **<< Dec Data. >>** ;

quit ;
```

Weather for SDG&E: Heating Degree Days – Average and Cold Year Designs; and Winter Peak Day Design Temperatures

September 2022

I. Overview

San Diego Gas and Electric Company's service area for natural gas extends from southern Orange County throughout San Diego County to the Mexican border. To quantify the overall temperature experienced within this region, SDG&E aggregates daily temperature recordings from three U.S. Weather Bureau weather stations into one system average heating degree-day ("HDD") figure. The table below lists weather station locations along with its associated temperature zone(s).

Table 1

Representative Weather Stations with Temperature Zones

Station Location	**Weight**	**Temperature Zone**
1. Miramar Naval Air Station[1]	1/3	Coastal and Inland
2. San Diego Lindbergh Field (International Airport)	(2/3) x (#Coastal/(#Coastal + #Inland))	Coastal
3. El Cajon	(2/3) x (#Inland/(#Coastal + #Inland))	Inland

SDG&E uses 65° Fahrenheit to calculate the number of HDDs. One heating degree-day is accumulated for each degree that the daily average is *below* 65° Fahrenheit. To arrive at the system average HDDs figure for its entire service area, SDG&E weights the HDD figure for each zone using the weights[2] shown in Table 1. These weights are used in calculating the data shown from January 2002 to December 2021.

Daily maximum and minimum temperatures, for each individual weather station in the table above, are compiled from National Weather Service data, at the website http://www.wrh.noaa.gov/sgx/obs/rtp/rtpmap.php?wfo=sgx which provides easy access to temperature data for San Diego and parts of

---

[1] The location of the station for Miramar is at the boundary of the Coastal and Inland zones. Correspondingly, both the Coastal and Inland zones are considered represented in the data for the Miramar station.

[2] As of December 2021, there were 488,783 gas customers associated with the Coastal temperature zone and 418,743 gas customers associated with the Inland temperature zone. The following URL shows a map of the SDG&E service area and temperature zones: http://www.sdge.com/tm2/pdf/ELEC_MAPS_Maps_-_Elec.pdf ; less than 0.04% of SDG&E's gas customers were in the mountain and desert zones.

surrounding counties. For each station, the average temperature is computed as the (maximum + minimum)/2 and this value is used to compute the heating degrees (i.e., the *daily* HDD) for each station as well. System average values of HDD are then computed using the weights for each respective station. Annual and monthly HDDs for the entire SDG&E service area from 2002 to 2021 are listed in Table 2, below.

Table 2

Calendar Month Heating Degree-Days (Jan. 2002 through Dec. 2021)

	Month												Total
Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	"Cal-Year"
2002	315	225	247	158	90	13	0	0	2	54	81	294	**1479**
2003	141	201	179	184	95	32	0	0	0	7	157	275	**1271**
2004	273	269	97	65	14	4	1	0	0	52	200	266	**1241**
2005	244	197	159	118	33	5	0	0	4	38	95	231	**1124**
2006	275	204	305	144	33	0	0	0	1	35	88	287	**1372**
2007	365	225	155	139	64	20	0	0	4	28	112	340	**1452**
2008	331	278	187	131	89	16	0	0	0	13	59	287	**1391**
2009	177	247	202	141	30	11	0	0	0	41	124	291	**1264**
2010	240	212	195	178	88	24	10	1	2	31	181	239	**1401**
2011	220	278	196	96	75	20	0	0	0	25	172	340	**1422**
2012	232	239	230	129	37	13	0	0	0	16	102	268	**1266**
2013	323	270	150	104	23	6	0	0	0	40	104	241	**1261**
2014	158	140	80	78	20	1	0	0	0	0	44	170	**691**
2015	161	87	58	44	46	0	0	0	0	0	105	259	**760**
2016	237	82	95	42	27	0	0	0	0	0	67	196	**746**
2017	243	156	82	27	42	4	0	0	0	1	38	149	**742**
2018	113	171	136	57	48	1	0	0	0	1	47	199	**773**
2019	227	301	173	59	72	4	0	0	2	13	94	241	**1186**
2020	230	193	176	94	3	0	0	0	0	6	132	242	**1076**
2021	250	191	231	91	42	5	0	0	1	48	72	305	**1236**
20-Yr-Avg (Jan2002-Dec2021)													
Avg.	237.8	208.3	166.7	104.0	48.6	9.0	0.6	0.1	0.8	22.5	103.7	256.0	**1157.7**
St.Dev.	65.8	60.3	62.9	46.3	27.9	9.3	2.2	0.2	1.3	19.0	46.1	50.8	**266.9**
Min.	113.0	82.0	58.0	27.0	3.0	0.0	0.0	0.0	0.0	0.0	38.0	149.0	**691.0**
Max.	365.0	301.0	305.0	184.0	95.0	32.0	10.0	1.0	4.0	54.0	200.0	340.0	**1479.0**

II. Calculations to Define Our Average-Temperature Year

The simple average of the 20-year period (January 2002 through December 2021) was used to represent the Average Year total and the individual monthly values for HDD. In this CGR, the standard deviation has been calculated using an approach that compensates for the annual HDD values for the years 2014-2018 in SDG&E's service territory being dramatically lower than in any preceding year going back to 1972. A regression with a time trend and a dummy variable for the years 2014-2018 has been used to estimate a shift in the level of annual HDD that occurred beginning in 2014. A dummy variable takes the value one for some observations to indicate the presence of an effect or membership in a group and zero for the remaining observations. Estimating the effect of the dummy variable gives an estimate of that effect or the impact of membership in that group. A dummy variable is used here to estimate the average effect on annual HDD of a given year having membership in the group of years 2014-2018. The dataset is SDG&E system-wide annual HDD for the years 2002-2021. The regression equation is:

$$HDD_t = \alpha + \beta * t + \beta_{2014-2018} * D_{2014-2018} + \varepsilon$$

where $D_{2014-2018}$ is a dummy variable for the years 2014-2018 and $\beta_{2014-2018}$ is the corresponding dummy coefficient. This regression equation estimates average HDD over the period 1998-2018 controlling for time trends in HDD and the warm weather regime of years 2002-2021. It's important to note that p-value for the estimate of $\beta_{2014-2018}$ is virtually zero, indicating an extremely low probability that membership in the group of years 2014-2018 had no effect on annual HDDs. Please see Table 3 below for the full regression output.

Table 3

Dummy Regression for Calculation of Heating Degree-Day Standard Deviation

Regression Statistics	
Multiple R	0.92161378
R Square	0.849371959
Adjusted R Squar	0.841003734
Standard Error	106.4348871
Observations	20

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	1149827.267	1149827.267	101.4996619	7.96345E-09
Residual	18	203910.9333	11328.38519		
Total	19	1353738.2			

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	1296.133333	27.48136967	47.16407329	0.000000000
Regime Dummy	-553.7333333	54.96273934	-10.07470406	0.000000008

The dummy variable's estimated effect, $\beta_{2014-2018}$, is subtracted from the actual annual HDD data for years 2014-2018 to adjust the data to remove the level shift. The standard deviation has been calculated to be 103.6 using this adjusted dataset. This adjusted standard deviation has been used to design the two Cold Years based on a "1-in-10" and "1-in-35" chance, ***c***, that the respective annual "Cold Year" $\boldsymbol{hdd_c}$ value would be exceeded. A probability model for the annual HDD is based on a t-Distribution with N-1 degrees of freedom, where N is the number of years of HDD data we use, μ is the average of the last 20 years of HDD, and S_{20} is the average of the standard deviations of the 20 most recent 20 year periods:

$U = (HDD_y - \mu)/S_{20}$, has a t-Distribution with N-1 degrees of freedom.

III. Calculating the Cold-Temperature Year Weather Designs

Cold Year HDD Weather Designs

For SDG&E, cold-temperature-year HDD weather designs are developed with a 1-in-35 year chance of occurrence. In terms of probabilities this can be expressed as the following for a "1-in-35" cold-year HDD value in equation 1 and a "1-in-10" cold-year HDD value in equation 2, with Annual HDD as the random variable:

(1) Prob { Annual HDD > "1-in-35" Cold-Yr HDD } = 1/35 = 0.0286

(2) Prob { Annual HDD > "1-in-10" Cold-Yr HDD } = 1/10 = 0.1000

An area of 0.0286 under one tail of the T-Distribution translates to 2.025 standard deviations *above* an average-year based on a t-statistic with 19 degrees of freedom. Using the standard deviation calculated as described earlier, 103.6 HDD, these equations yield values of about 1,368 HDD for a "1-in-35" cold year and 1,296 as the number of HDDs for a "1-in-10" cold year (an area of 0.1000 under one tail of the T-Distribution translates to 1.328 standard deviations *above* an average-year based on a t-statistic with 19 degrees of freedom). For example, the "1-in-35" cold-year HDD is calculated as follows:

(3) Cold-year HDD = 1,368 which equals approximately

1,158 average-year HDDs + 2.025 * 103.6

Table 4 below shows monthly HDD figures for "1-in-35" cold year, "1-in-10" cold year and, average year temperature designs. The monthly average-temperature-year HDDs are calculated from weighted monthly HDDs from 2002 to 2021, as shown as the bottom of Table 2, above. For example, the average-year December value of 256.0 HDD equals the simple average of the 20 December HDD figures from 2002 to 2021. SDG&E calculates the cold--temperature-year monthly HDD values using the same shape of the average-year HDDs. For example, since 22.1 percent (256.0 / 1157.7) of average-temperature-year HDDs occurred in December, the estimated number of HDDs during December for a cold-year is equal to 1,368 HDDs multiplied by 22.1 percent, or 302.5 HDDs.

Table 4

Calendar Month Heating Degree-Day Designs

	Cold		Average	Hot	
	1-in-35 Design	1-in-10 Design		1-in-10 Design	1-in-35 Design
January	280.9	266.2	237.8	209.5	194.7
February	246.1	233.2	208.4	183.5	170.6
March	196.9	186.6	166.7	146.8	136.5
April	122.8	116.4	104.0	91.6	85.1
May	57.4	54.3	48.6	42.8	39.8
June	10.6	10.0	9.0	7.9	7.3
July	0.6	0.6	0.6	0.5	0.5
August	0.1	0.1	0.1	0.0	0.0
September	0.9	0.9	0.8	0.7	0.7
October	26.5	25.1	22.5	19.8	18.4
November	122.5	116.1	103.7	91.4	84.9
December	302.5	286.6	256.1	225.6	209.6
	1368	1296	1158	1020	948

IV. Adjusting Forecasted HDDs for a Climate-Change Trend

SDG&E incorporates a climate-change warming trend that gradually reduces HDDs by 6 HDDs per year over the forecast period. The annual reduction is based on the latest twenty-year trend in 20-year-averaged HDDs. That is, they are based on the observed trend in changes starting with average HDDs for years 1983-2002, then 1984-2003, 1985-2004...and ending with the average HDDs for years 2002-2021.

Table 5 below shows system HDDs, rolling 20-year averaged HDDs, and the annual changes in those rolling 20-year averages. The actual average annual change is -5.9 HDDs for the most recent twenty of the 20-year averages (with ending years from 2002 through 2021). A simple “ordinary least squares” regression-fitted time trend (using Microsoft Excel’s “LINEST” function) was applied to those same annual changes, resulting in a fitted estimation of -6.2 HDDs per year. Given the fitted and actual average annual changes of -6.2 and -5.9 HDDs, respectively, it was decided to decrease average-year and cold-year forecasted HDD’s by an even 6 HDDs per year, starting with the first forecast year of 2022.

Table 5

Average Annual Changes in 20-Year Averaged Heating-Degree Days

Average Annual Changes in 20-Year-Averaged HDDs		
	Regression Fitted trend	Actual
20 years (2002-2021)	-6.2	-5.9

Year	SDG&E System HDDs	20-year averaged HDDs	Annual change in 20-year averaged HDDs
1982	1332		
1983	1106		
1984	1115		
1985	1395		
1986	1028		
1987	1400		
1988	1269		
1989	1259		
1990	1316		
1991	1317		
1992	994		
1993	1108		
1994	1464		
1995	1073		
1996	1151		
1997	1152		
1998	1571		
1999	1610		
2000	1326		
2001	1543	1276.5	
2002	1479	1283.8	7.3
2003	1271	1292.1	8.3
2004	1241	1298.4	6.3
2005	1124	1284.8	-13.6
2006	1372	1302.0	17.2
2007	1452	1304.6	2.6
2008	1391	1310.7	6.1
2009	1264	1311.0	0.3
2010	1401	1315.2	4.3
2011	1422	1320.5	5.3
2012	1266	1334.1	13.6
2013	1261	1341.7	7.7
2014	691	1303.1	-38.7
2015	760	1287.4	-15.6
2016	746	1267.2	-20.3
2017	742	1246.7	-20.5
2018	773	1206.8	-39.9
2019	1186	1185.6	-21.2
2020	1076	1173.1	-12.5
2021	1236	1157.7	-15.3

V. Calculating the Peak-Day Design Temperature

SDG&E's Peak-Day design temperature of 43.3 degrees Fahrenheit, denoted "Deg-F," is determined from a statistical analysis of observed annual minimum daily system average temperatures constructed from daily temperature recordings from the three U.S. Weather Bureau weather stations discussed above. Since we have a time series of daily data by year, the following notation will be used for the remainder of this discussion:

(1) $AVG_{y,d}$ = system average value of Temperature
for calendar year "y" and day "d".

The calendar year, y, can range from 1972 through 2021, while the day, d, can range from 1 to 365, for non leap years, or from 1 to 366 for leap years. The "upper" value for the day, d, thus depends on the calendar year, y, and will be denoted by n(y)=365, or 366, respectively, when y is a non-leap year or a leap year.

For each calendar year, we calculate the following statistic from our series of daily system average temperatures defined in equation (1) above:

(2) $$MinAVG_y = \min_{d=1}^{n(y)}\{ AVG_{y,d} \}, \text{ for } y=1972, 1973, \ldots, 2021.$$

(The notation used in equation 2 means "For a particular year, y, list all the daily values of system average temperature for that year, then pick the smallest one.")

The resulting minimum annual temperatures are shown in Table 6, below. Most of the minimum temperatures occur in the months of December, January, or February; for a few calendar years the minimums occurred in March or November.

The statistical methods we use to analyze this data employ software developed to fit three generic probability models: the Generalized Extreme Value (GEV) model, the Double-Exponential or GUMBEL (EV1) model and a 2-Parameter Students' T-Distribution (T-Dist) model. [The GEV and EV1 models have the same mathematical specification as those implemented in a DOS-based executable-only computer code that was developed by Richard L. Lehman and described in a paper published in the Proceedings of the Eighth Conference on Applied Climatology, January 17-22, 1993, Anaheim, California, pp. 270-273, by the American Meteorological Society, Boston, MA., with the title "Two Software Products for Extreme Value Analysis: System Overviews of ANYEX and DDEX." At the time he wrote the paper, Dr. Lehman was with the Climate Analysis Center, National Weather Service/NOAA in Washington, D.C., zip code 20233.] The Statistical Analysis Software (SAS) procedure for nonlinear statistical model estimation (PROC MODEL, from SAS V6.12) was used to do the calculations. Further, the calculation procedures were implemented to fit the probability models to observed *maximums* of data, like heating degrees. By recognizing that:

n(y) n(y)

$$- \text{MinAVG}_y = - \min_{d=1}\{\text{AVG}_{y,d}\} = \max_{d=1}\{ -\text{AVG}_{y,d}\}, \text{ for } y=1972, \ldots, 2021;$$

this same software, when applied to the *negative* of the minimum temperature data, yields appropriate probability model estimation results.

The calculations done to fit any one of the three probability models chooses the parameter values that provide the "best fit" of the parametric probability model's calculated cumulative distribution function (CDF) to the empirical cumulative distribution function (ECDF). Note that the ECDF is constructed based on the variable "-MinAVG_y" (which is a *maximum* over a set of *negative* temperatures) with values of the variable MinAVG_y that are the same as shown in Table 6, below.

In Table 6, the data for -MinAVG_y are shown after they have been sorted from "lowest" to "highest" value. The ascending *ordinal* value is shown in the column labeled "RANK" and the empirical cumulative distribution function is calculated and shown in the next column. The formula used to calculate this function is:

$$\text{ECDF} = (\text{RANK} - \alpha)/[\text{MaxRANK} + (1 - 2\,\alpha)],$$

where the parameter "α" (shown as *alpha* in Table 7) is a "small" positive value (usually less than ½) that is used to bound the ECDF away from 0 and 1.

Of the three probability models considered (GEV, EV1, and T_Dist) the results obtained for the T_Dist model were selected since the fit to the ECDF was better than that of either the GEV model or the EV1 model. (Although convergence to stable parameter estimates is occasionally a problem with fitting a GEV model to the ECDF, the T_Dist model had no problems with convergence of the iterative procedure to estimate parameters.)

The T_Dist model used here is a three-parameter probability model where the varable $z = (-\text{MinAVG}_y - \gamma) / \theta$, for each year, y, is presumed to follow a T_Dist with location parameter, γ, and scale parameter, θ, and a third parameter, v, that represents the number of degrees of freedom. For a given number of years of data, N, then v=N-2.

The following mathematical expression specifies the T_Dist model we fit to the data for "-MinAVG_y " shown in Table 6, below.

(3) $\text{ECDF}(-\text{MinAVG}_y) = \text{Prob} \{ -T < -\text{MinAVG}_y \} = \text{T\_Dist}\{z; \gamma, \theta, v=N-2\}$,

where "T_Dist{ . }" is the cumulative probability distribution function for Student's T-Distribution[3], and

[3] A common mathematical expression for Student's T-Distribution is provided at http://en.wikipedia.org/wiki/Student%27s_t-distribution; with a probability density function

$$f(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\,\Gamma(\frac{\nu}{2})}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

such that T_Dist{z; γ, θ, ν=N-2}=∫f(t) dt, from t=-∞ to t=z. Also, the notation Γ(.) is known in mathematics as the GAMMA function; see http://www.wikipedia.org/wiki/Gamma_function for a description. Also, see *Statistical Theory*, 3rd Ed., B.W. Lindgren, MacMillian Pub. Inc, 1976, pp. 336-337.

(4) $z = (-MinAVG_y - \gamma) / \theta$, for each year, y, and

the parameters "γ" and "θ" are estimated for this model for given degrees of freedom v=N-2. The estimated values for γ and θ are shown in Table 7 along with the fitted values of the model CDF (the column: "Fitted" Model CDF).

Now, to calculate a *peak-day design temperature*, $TPDD_\delta$, with a specified likelihood, δ, that a value less than $TPDD_\delta$ would be observed, we use the equation below:

(5) $\delta = Prob \{ T \leq TPDD_\delta \}$, which is equivalent to

(6) $\delta = Prob \{ [(-T - \gamma) / \theta] \geq [(-TPDD_\delta - \gamma) / \theta] \}, = Prob \{ [(-T - \gamma) / \theta] \geq [z_\delta] \}$,

where $z_\delta = [(-TPDD_\delta - \gamma) / \theta]$. In terms of our probability model,

(7) $\delta = 1 - T\_Dist\{ z_\delta; \gamma, \theta, v=N-2\}$,

which yields the following equation for z_δ,

(7') $z_\delta = \{ TINV\_Dist\{ (1-\delta); \gamma, \theta, v=N-2\}$, where "TINV_Dist{ . }" is the inverse function of the T_Dist{ . } function[4]. The implied equation for $TPDD_\delta$ is:

(8) $TPDD_\delta = - [\gamma + (z_\delta)(\theta)]$.

To calculate the minimum daily (system average) temperature to define our extreme weather event, we specify that this COLDEST-Day be one where the temperature would be lower with a "1-in-35" likelihood. This criterion translates into two equations to be solved based on equations (7) and (8) above:

(9) solve for "z_δ" from equation (7') above with (1-δ) = (1 - 1/35) = 1 - 0.0286,

(10) solve for "$TPDD_\delta$" from $TPDD_\delta = - [\gamma + (z_\delta)(\theta)]$.

The value of $z_\delta = 1.951$ and $TPDD_\delta = - [\gamma + (z_\delta)(\theta)] = 43.3$ degrees Fahrenheit, with values for "v=N-2"; along with "γ" and "θ" in Table 7, below.

SDG&E's "1-in-10" peak-day design temperature of 44.8 degrees Fahrenheit, is calculated in a methodologically similar way as for the 43.3 degree "1-in-35" peak day temperature. The criteria specified in equation (9) above for a "1-in-35" likelihood would be replaced by a "1-in-10" likelihood.

(9') solve for "z_δ" from equation (7') above with (1-δ) = (1 - 1/10) = 1 - 0.1000,

which yields a "z_δ" value of $z_\delta = 1.300$ and, $TPDD_\delta = - [\gamma + (z_\delta)(\theta)] = 44.6$ with values for "v=N-2"; along with "γ" and "θ" in Table 7, below.

A plot of the cumulative distribution function for $MinAVG_y$ based on "v=N-2", the fitted model parameters, "γ" and "θ" with values in Table 7, below, is shown in Figure 1.

Table 6

[4] Computer software packages such as SAS and EXCEL have implemented statistical and mathematical functions to readily calculate values for T_Dist{ . } and TINV_Dist{ . } as defined above.

YEAR	MINAVG	Month(MinAvg)
1972	46.7634	Dec
1973	46.1862	Jan
1974	44.2058	Dec
1975	44.1862	Jan
1976	45.0710	Jan
1977	50.6605	Mar
1978	42.7120	Dec
1979	45.1544	Jan
1980	53.7953	Jan
1981	49.8467	Jan
1982	48.8210	Dec
1983	51.4877	Jan
1984	48.4620	Dec
1985	46.1029	Dec
1986	50.1029	Feb
1987	41.4877	Dec
1988	45.4362	Dec
1989	45.1544	Jan
1990	43.7696	Feb
1991	48.7696	Mar
1992	47.1544	Dec
1993	46.7696	Jan
1994	48.0515	Nov
1995	51.1544	Dec
1996	48.7696	Feb
1997	49.0772	Dec
1998	46.7696	Dec
1999	48.7953	Jan
2000	50.3591	Jan
2001	47.6924	Jan
2002	45.7438	Jan
2003	49.0515	Dec
2004	47.7438	Nov
2005	47.7953	Jan
2006	48.3591	Dec
2007	43.3591	Jan
2008	48.7181	Dec
2009	48.4105	Feb
2010	48.1801	Dec
2011	49.0772	Feb
2012	48.1286	Dec
2013	44.1286	Jan
2014	47.7696	Dec
2015	48.2596	Jan
2016	50.2981	Feb
2017	51.2670	Jan
2018	49.9341	Dec
2019	48.4877	Feb
2020	48.1029	Feb
2021	47.6409	Dec

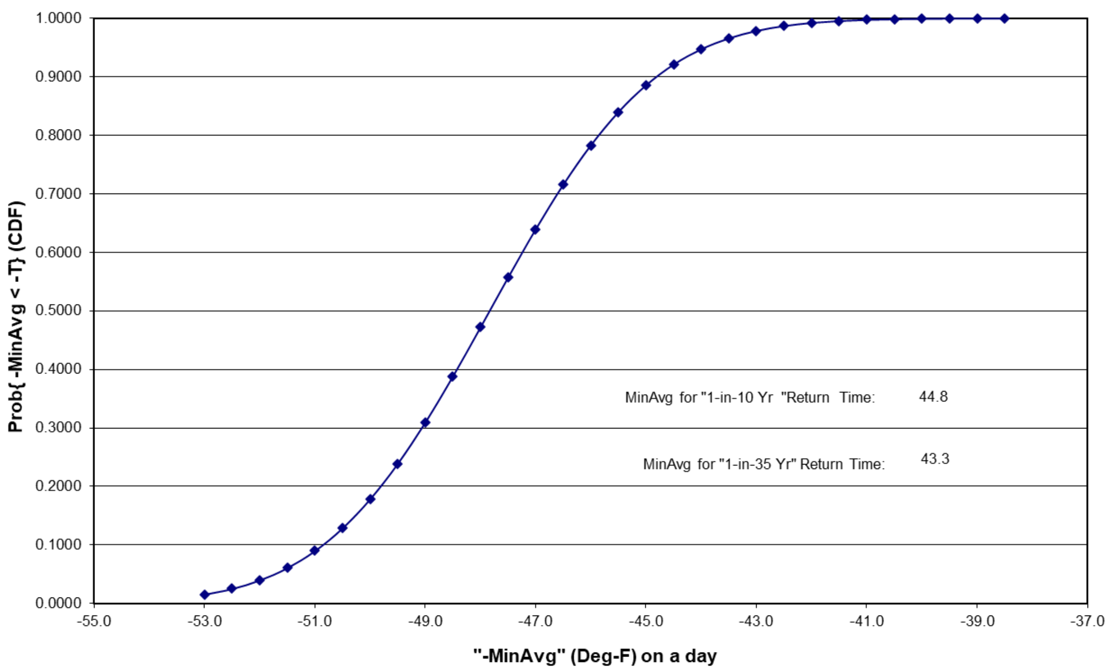
Table 7 (alpha=0.375)

Year	Days/Yr	-MinAvg	Month (-MinAvg)	Rank	Emprical CDF	Fitted Model CDF
1980	366	-53.7953	Jan	1	0.0124	-2.4766
1983	365	-51.4877	Jan	2	0.0323	-1.5444
2017	365	-51.2670	Jan	3	0.0522	-1.4553
1995	365	-51.1544	Dec	4	0.0721	-1.4098
1977	365	-50.6605	Mar	5	0.0920	-1.2103
2000	366	-50.3591	Jan	6	0.1119	-1.0885
2016	366	-50.2981	Feb	7	0.1318	-1.0639
1986	365	-50.1029	Feb	8	0.1517	-0.9850
2018	365	-49.9341	Dec	9	0.1716	-0.9169
1981	365	-49.8467	Jan	10	0.1915	-0.8816
1997	365	-49.0772	Dec	11	0.2114	-0.5707
2011	365	-49.0772	Feb	12	0.2313	-0.5707
2003	365	-49.0515	Dec	13	0.2512	-0.5603
1982	365	-48.8210	Dec	14	0.2711	-0.4672
1999	365	-48.7953	Jan	15	0.2910	-0.4568
1991	365	-48.7696	Mar	16	0.3109	-0.4464
1996	366	-48.7696	Feb	17	0.3308	-0.4464
2008	366	-48.7181	Dec	18	0.3507	-0.4256
2019	365	-48.4877	Feb	19	0.3706	-0.3325
1984	366	-48.4620	Dec	20	0.3905	-0.3222
2009	365	-48.4105	Feb	21	0.4104	-0.3014
2006	365	-48.3591	Dec	22	0.4303	-0.2806
2015	365	-48.2596	Jan	23	0.4502	-0.2404
2010	365	-48.1801	Dec	24	0.4701	-0.2083
2012	366	-48.1286	Dec	25	0.4900	-0.1875
2020	366	-48.1029	Feb	26	0.5100	-0.1771
1994	365	-48.0515	Nov	27	0.5299	-0.1563
2005	365	-47.7953	Jan	28	0.5498	-0.0528
2014	365	-47.7696	Dec	29	0.5697	-0.0425
2004	366	-47.7438	Nov	30	0.5896	-0.0321
2001	365	-47.6924	Jan	31	0.6095	-0.0113
2021	365	-47.6409	Dec	32	0.6294	0.0095
1992	366	-47.1544	Dec	33	0.6493	0.2061
1993	365	-46.7696	Jan	34	0.6692	0.3615
1998	365	-46.7696	Dec	35	0.6891	0.3615
1972	366	-46.7634	Dec	36	0.7090	0.3640
1973	365	-46.1862	Jan	37	0.7289	0.5972
1985	365	-46.1029	Dec	38	0.7488	0.6308
2002	365	-45.7438	Jan	39	0.7687	0.7759
1988	366	-45.4362	Dec	40	0.7886	0.9001
1979	365	-45.1544	Jan	41	0.8085	1.0140
1989	365	-45.1544	Jan	42	0.8284	1.0140
1976	366	-45.0710	Jan	43	0.8483	1.0477
1974	365	-44.2058	Dec	44	0.8682	1.3972
1975	365	-44.1862	Jan	45	0.8881	1.4051
2013	365	-44.1286	Jan	46	0.9080	1.4283
1990	365	-43.7696	Feb	47	0.9279	1.5734
2007	365	-43.3591	Jan	48	0.9478	1.7392
1978	365	-42.7120	Dec	49	0.9677	2.0006
1987	365	-41.4877	Dec	50	0.9876	2.4952

"Gamma" (Fitted) =	**-47.84**
"Theta" (Fitted) =	**2.33**
Deg. Freedom=	**48**

Figure 1

CDF for the Random Variable: "-MinAvg",
[Minimum System Avg. Temp (Deg-F) on a Day over a Year]



VI. Estimating the Uncertainty in the Peak-Day Design Temperature

The calculated peak-day design temperatures in section V above also have a statistical uncertainty associated with them. The estimated measures of uncertainty recommended for our use are calculated from the fitted model for the probability distribution and are believed to be reasonable, although rough, approximations.

The basic approach used the estimated parameters for the probability distribution (see the results provided in Table 7, above) to calculate the fitted temperatures as a function of the empirical CDF listed in Table 7. These fitted temperatures are then "compared" with the observed temperatures by calculating the difference = "observed" – "fitted" values. The full set of differences are then separated into the lower third (L), the middle third (M) and the upper third (U) of the distribution. Finally, calculate values of the root-mean-square error (RMSE) of the differences in each third of the distribution, along with the entire set of differences overall. The data in Table 8, below, show the temperature data and the resulting RMSE values.

The formula below is used to calculate the RMSE for a specified set of "N" data differences:

$$\text{RMSE} = \text{SQRT}\left\{ \left(\sum_{i=1, \ldots, N} e[\,i\,]^{2} \right) / (N-2) \right\},$$

where e[i] = *observed* less *fitted* value of temperature, T[i]. The number of estimated parameters (3 for the GEV model, 2 for the T-Dist and EV1 models) is subtracted from the respective number of data differences, N, in the denominator of the RMSE expression.

Since both the "1-in-35" and "1-in-10" peak-day temperature values are in the lower third quantile of the fitted distribution, the calculated standard error for these estimates is 0.60 Deg-F.

Table 8

Quantile: (Lower, Middle, Upper 3rd's)	Observed $T_{[i]}$ Temp. Ranked	Fitted Value of $T_{[i]}$	Residual $e_{[i]}$: Obs'd. less Fitted Value of $T_{[i]}$	Square of $e_{[i]}$:	
U	53.7953	53.2170	0.5783	0.334375	
U	51.4877	52.2310	-0.7434	0.552571	
U	51.2670	51.6830	-0.4160	0.173093	
U	51.1544	51.2868	-0.1324	0.017542	
U	50.6605	50.9696	-0.3091	0.095533	
U	50.3591	50.7012	-0.3421	0.117054	
U	50.2981	50.4660	-0.1679	0.028198	
U	50.1029	50.2549	-0.1520	0.023099	
U	49.9341	50.0620	-0.1278	0.016343	
U	49.8467	49.8833	-0.0366	0.001337	
U	49.0772	49.7160	-0.6388	0.408110	
U	49.0772	49.5580	-0.4808	0.231165	
U	49.0515	49.4076	-0.3561	0.126823	
U	48.8210	49.2635	-0.4425	0.195823	
U	48.7953	49.1248	-0.3295	0.108602	
U	48.7696	48.9906	-0.2211	0.048875	
U	48.7696	48.8602	-0.0907	0.008221	
M	48.7181	48.7330	-0.0149	0.000222	
M	48.4877	48.6085	-0.1208	0.014592	
M	48.4620	48.4862	-0.0242	0.000585	
M	48.4105	48.3656	0.0449	0.002014	
M	48.3591	48.2465	0.1125	0.012660	
M	48.2596	48.1285	0.1310	0.017169	
M	48.1801	48.0113	0.1688	0.028488	
M	48.1286	47.8945	0.2341	0.054813	
M	48.1029	47.7779	0.3250	0.105649	
M	48.0515	47.6611	0.3904	0.152392	
M	47.7953	47.5438	0.2515	0.063232	
M	47.7696	47.4258	0.3437	0.118156	
M	47.7438	47.3067	0.4371	0.191060	
M	47.6924	47.1862	0.5062	0.256214	
M	47.6409	47.0639	0.5771	0.332990	
M	47.1544	46.9393	0.2150	0.046228	
L	46.7696	46.8121	-0.0426	0.001812	
L	46.7696	46.6817	0.0878	0.007716	
L	46.7634	46.5475	0.2159	0.046606	
L	46.1862	46.4088	-0.2226	0.049550	
L	46.1029	46.2648	-0.1619	0.026210	
L	45.7438	46.1144	-0.3706	0.137311	
L	45.4362	45.9564	-0.5201	0.270530	
L	45.1544	45.7891	-0.6347	0.402854	
L	45.1544	45.6104	-0.4560	0.207967	
L	45.0710	45.4175	-0.3465	0.120038	
L	44.2058	45.2064	-1.0006	1.001128	
L	44.1862	44.9712	-0.7849	0.616140	
L	44.1286	44.7028	-0.5741	0.329648	
L	43.7696	44.3856	-0.6160	0.379461	
L	43.3591	43.9894	-0.6303	0.397285	
L	42.7120	43.4413	-0.7294	0.531984	
L	41.4877	42.4553	-0.9676	0.936331	
			Overall RMSE ($e_{[i]}$):	**0.43**	**°F**
			Upper 3rd RMSE ($e_{[i]}$):	**0.41**	**°F**
			Middle 3rd RMSE ($e_{[i]}$):	**0.32**	**°F**
			Lower 3rd RMSE ($e_{[i]}$):	**0.60**	**°F**

VII. The Relationship between Annual Likelihoods for Peak-Day Temperatures and "Expected Return Time"

The event whose probability distribution we've modeled is the likelihood that the minimum daily temperature over a calendar year is less than a specified value. And, in particular, we've used this probability model to infer the value of a temperature, our *peak-day design temperature* ($TPDD_\delta$), that corresponds to a pre-defined likelihood, δ, that the observed minimum temperature is less than or equal to this design temperature.

(1) δ = Prob{ Minimum Daily Temperature over the Year < $TPDD_\delta$ }.

For some applications, it is useful to think of how this specified likelihood (or "risk level" δ) relates to the expected number of years until this Peak-Day event would first occur. This expected number of years is what is meant by the *return period*. The results stated below are found in the book: ***Statistics of Extremes***, E.J. Gumbel, Columbia University Press, 1958, on pages 21-25.

(2) E[#Yrs for Peak-Day Event to Occur] = 1 / δ,

1 / Prob{ Minimum Daily Temperature over the Year < $TPDD_\delta$ }.

For our peak-day design temperature (43.3°F) associated with a 1-in-35 annual likelihood, the return period is 35 years (δ=1/35). For the 44.8°F peak-day design temperature, the return period is 10 years (δ=1/10). Occasionally, a less precise terminology is used. For example, the 43.3°F peak-day design temperature may be referred to as a "1-in-35 year cold day"; and the 44.8°F peak-day design temperature may be referred to as a "1-in-10 year cold day."

The probability model for the *return period*, as a random variable, is a geometric (discrete) distribution with positive integer values for the *return period*. The parameter δ = Prob{ Minimum Daily Temperature over the Year < $TPDD_\delta$ }.

(3) Prob{ *return period* = r } = $(1 - \delta)^{(r-1)}\,\delta$, for r = 1, 2, 3, . . .

The expected value of the *return period* is already given in (2) above; the variance of the *return period* is:

(4) Var[*return period*] = (E[*return period*])^2 x (1- (1 / E [*return period*])),

(4') Var[*return period*] = (E[*return period*]) x (E [*return period*] - 1).

Equations (4) and (4') indicate that the standard deviation (square root of the variance) of the *return period* is nearly equal to its expected value. Thus, there is substantial variability about the expected value—a *return period* is not very precise.

SDG&E SAS Code #1: SysAvTmp(SCG-Method)_CustCntWgts-v2.sas

```
title1 "Calculation of 'System Average' Temperature for SDGandEs Gas System Load";
title2 "(Weighting of three stations: El Cajon, San Diego (Lindbergh Field) and Miramar Naval Air Station)" ;


*****************************************************************************************************;
****                                                                    ****;
****   [System-wide daily temperature series: ]
****;
****                                                                    ****;
****   SECOND Alternative Weighting Method (Alt2)                                      ****;
****                                                                    ****;
****                                                                    ****;
*****************************************************************************************************;



options date number source notes ;

options mprint ;
/* %cour8l
   %cour8p */

options nomprint;

**<<LANDSCAPE: Set Line and Page Size for "SAS-Monospace and Roman 6pt.>>**;
options ls=211 ps=69;

**<<PORTRAIT: Set Line and Page Size for "SAS-Monospace and Roman 6pt.>>**;
**options ls=160 ps=90;

options date number source notes ;



****<< Directory for "SDWeathr" worksheet file:                          >>****;
****<< "S:\SDGandE-Econometric-Models\Weather-Data\Cycle\Sdge-Cycle-Methodology\"        >>****;



**<< Weather data file now maintained by SDGandE's Electric Demand Forecasting group.          >> **;
**<<   URL for SDG&E's data: "http://strategicanalysisd.sempra.com/Weather/Default.aspx"        >> **;


libname lb2   '/EDS_RB/Weather/2022 CGR/SDGE' ; ***<< Directories for 2022 CGR  weather data
analsysis. >>***;
libname LbDly '/EDS_RB/Weather/2022 CGR/SDGE' ;

proc import out = SDWeathr
          datafile = "/EDS_RB/Weather/2022 CGR/SDGE/SanDiegoWeather-20220318-MODIFIED.xls"
          dbms = xls replace;
                    sheet = "Temp";
                    getnames = yes;
run;
```

```
data DailySD ;
            set SDWeathr;

   *infile "/data/home/scgdf/2016_CGR_Weather/SDGE/sdweathr.sas7bdat" ;
   *** informat Date date12.   S42706Mx   S42706Mn   S47740Mx   S47740Mn   S93107Mx   S93107Mn
6.2 ;
   ***   << Format above is for "WthrbkInfo.xls" or "Wthrbkinfo-Rev.xls"  DATE variable!  >>  *** ;

   *informat Date mmddyy10.   S42706Mx   S42706Mn   S47740Mx   S47740Mn   S93107Mx   S93107Mn
6. ;

    ***<<< Note: For S# for El Cajon is 42706=CoopId# from NWS Weather Station Id# list, and
>>>*** ;
    ***<<<       the S# for San Diego, Lindberg Field is 47740=CoopId# from NWS Weather Station Id# list.
>>>*** ;
    ***<<<    But, the S# for Miramar, Naval Air Station is 93107=WBanID# from NWS Weather Station Id#
list. >>** ;

*   input Date      S42706Mx   S42706Mn   S47740Mx   S47740Mn   S93107Mx   S93107Mn  ;
*   format Date mmddyy8. ;

   if (date >= mdy(01,01,1972)
       AND date <= mdy(03,15,2022)) ; **<< Select only dates through December 31, 2021  >>**;

   month = month(date) ;
   year = year(date) ;

   label S42706Mx  ='El Cajon - Max';
   label S42706Mn  ='El Cajon - Min';
   label S93107Mx  ='Miramar NAS - Max';
   label S93107Mn  ='Miramar NAS - Min';
   label S47740Mx  ='S.D. (Lindbergh Fld.) - Max';
   label S47740Mn  ='S.D. (Lindbergh Fld.) - Min';

   label avg = "Syst-Avg. Avg" ;
   label max = "Syst-Avg. Max" ;
   label min = "Syst-Avg. Min" ;
   label hdd = "Syst-Avg. Hdd" ;
   label cdd = "Syst-Avg. Cdd" ;


     year  = year(date);
     month = month(date);
     ARRAY A_Mx(I) S42706Mx   S47740Mx    S93107Mx  ;
     ARRAY A_Mn(I) S42706Mn   S47740Mn    S93107Mn  ;
     ARRAY A_Av(I) S42706Av   S47740Av    S93107Av  ;
     ARRAY A_Hd(I) S42706Hd   S47740Hd    S93107Hd  ;
     ARRAY A_Cd(I) S42706Cd   S47740Cd    S93107Cd  ;

     DO OVER A_Mx;
        A_Av = (A_Mx+A_Mn)/2;
       IF A_Av-INT(A_Av)=.5 THEN IF MOD(INT(A_Av),2)=0
       THEN A_Av=INT(A_Av);
       ELSE A_Av=INT(A_Av)+1;
       a_hd = max(0,65-a_av);
       a_cd = max(0,a_av-65);
     END;

    ***<<<  The equations below were used initially, but they are not exactly the correct method we have
>>>*** ;
    ***<<<  used with the SoCalGas temperature data anslysis work.                              >>>*** ;
    ***Avg = 0.35*(S42706Mx   + S42706Mn  ) + 0.30*(S47740Mx   + S47740Mn  )
```

```
          + 0.35*(S93107Mx   +  S93107Mn  ) ;
   ***Avg = Avg/2 ;    **<< Do not round! >>** ;
   ***Hdd = max(0,65-Avg) ;  **<< Do not round! >>** ;
   ***Cdd = max(0,Avg-65) ;  **<< Do not round! >>** ;


/*****************************************
   ***<<< The equation below for "Avg" is the one we used perviously! >>>*** ;

   Avg = 0.35*( S42706Av ) + 0.30*( S47740Av )
          + 0.35*( S93107Av ) ;
   Max = 0.35*( S42706Mx ) + 0.30*( S47740Mx )
          + 0.35*(S93107Mx ) ;
   Min = 0.35*( S42706Mn ) + 0.30*( S47740Mn )
          + 0.35*( S93107Mn ) ;
*****************************************/



/*****************************************
*****************************************/
 ***<<<                                                          >>>*** ;
 ***<<< The equation below for "Avg" is the one we NOW use.                      >>>*** ;
 ***<<<                                                          >>>*** ;
 ***<<<    1). GKK indicated recently (per his e-mail on 7/31/2007) at 12:09 PM,      >>>*** ;
 ***<<<       that he had changed the station weights to the same (1/3)             >>>*** ;
 ***<<<       for each station.                                       >>>*** ;
 ***<<<       (Apparently, he did this "some time ago"!!!)                    >>>*** ;
 ***<<<    2). BMW believes this SHOULD HAVE been done a LOT sooner.                 >>>*** ;
 ***<<<       The prior weighting always seemed to be arbitrary and could           >>>*** ;
 ***<<<       not be documented as being based on customer/meter count             >>>*** ;
 ***<<<       data.                                                 >>>*** ;
 ***<<<    3). BMW believes this equal weighting for each station will be           >>>*** ;
 ***<<<       more in line with the six-zone appraoch we use for SoCalGas.          >>>*** ;
 ***<<<       With "system integration" on the horizon, we would then             >>>*** ;
 ***<<<       treat SDG&E as simply a "7th" temperature zone, with a seventh        >>>*** ;
 ***<<<       zone-weight based on the number of SDG&E active residential          >>>*** ;
 ***<<<       meter/customer count that is around  804,545 for June 2007           >>>*** ;
 ***<<<       from Al Burye "R2Page" gas data.                               >>>*** ;
 ***<<<                                                          >>>*** ;


 **<< Equal wgts for each station. >>** ;
 ** Avg =    (( S42706Av ) + ( S47740Av ) +  ( S93107Av )) / 3 ;
 ** Max =    (( S42706Mx ) + ( S47740Mx ) +  ( S93107Mx )) / 3 ;
 ** Min =    (( S42706Mn ) + ( S47740Mn ) +  ( S93107Mn )) / 3 ;

 ** do over a_hd ;  ***<<< Round Degree-Days for each station. >>>*** ;
 **    a_hd = round(a_hd) ;
 **    a_cd = round(a_cd) ;
 ** end ;

     **<< Equal wgts for each station. >>** ;
 ** Hdd =    (( S42706Hd ) + ( S47740Hd ) +  ( S93107Hd )) / 3 ;    **<< Do not round! >>** ;
 ** Cdd =    (( S42706Cd ) + ( S47740Cd ) +  ( S93107Cd )) / 3 ;    **<< Do not round! >>** ;


/**** Per Downie Beckett's e-mail of February 2, 2015  ***************

(We can assume all the SDGE's customers with gas services also have electric services.
Here is the count for gas services by climate zone.
Representative for December-2014, billing month.
```

```
ClimateZone    Gas Service Points      Electric Service Points (including the gas customers)

Coastal      469,087                    806,105
Mountain         12                     17,644
Desert          313                     3,584
Inland      398,037                     589,279
*******************************************************************/


 **<< SECOND Alternative Weighting Method (Alt2)            >>** ;
 **<< Weighted using Coastal & Inland counts ONLY.          >>** ;
 **<<                                           >>** ;
 **<< Lindberg Field    ---->  Coastal (SdTz1)             >>** ;
 **<< El Cajon (Santee) ---->  Inland  (SdTz2)             >>** ;
 **<< Miramar           ---->  Coastal + Inland            >>** ;

 **<<  Based on analysis of this and method "Alt1", it makes the better        >>** ;
 **<<  transition from the simple "1/3" weighting for each location to one      >>** ;
 **<<  based on current SDGE gas-customer count data. (BMW, 4/8/2011.}         >>** ;
 **<<  (See work done at "S:\Weather\2010Cgr\SDGandE-Alt1Wgt\"                >>** ;
 **<<   and "S:\Weather\2010Cgr\SDGandE-Alt2Wgt\")                        >>** ;
 **<<                                                          >>** ;
 **<<  Formula:                                                    >>** ;
 **<<     Sys-Avg = (1/3) * (Miramar)                                     >>** ;
 **<<            [+ (1/3) * (Inland_Cnt)  * (El-Cajon/Santee)               >>** ;
 **<<             + (1/3) * (Coastal_Cnt) * (Lindberg Field)]               >>** ;
 **<<                 / [Inland_Cnt + Coastal_Cnt]                        >>** ;
 **<<                                                          >>** ;

 Coastal = 488783 ;    **<< Per Jan-19-2022 data email from David Okuni        >>** ;
 Inland  = 418743 ;


   label S42706Mx  ='El Cajon - Max';
   label S42706Mn  ='El Cajon - Min';
   label S93107Mx  ='Miramar NAS - Max'; **<< It turns out that Miramar NAS is      >>** ;
   label S93107Mn  ='Miramar NAS - Min'; **<<  on the border of SdTz1 and SdTz2.    >>** ;
   label S47740Mx  ='S.D. (Lindberg Fld.) - Max';
   label S47740Mn  ='S.D. (Lindberg Fld.) - Min';

 SdTz1Mx = (1*S93107Mx + 2*S47740Mx) / (1+2) ;  **<< Coastal (SdTz1):  Miramar NAS & Lindberg Fld.
>>** ;
 SdTz2Mx = (1*S93107Mx + 2*S42706Mx) / (1+2) ;  **<< Inland  (SdTz2):  Miramar NAS & El Cajon(Santee)
>>** ;

 SdTz1Mn = (1*S93107Mn + 2*S47740Mn) / (1+2) ;  **<< Coastal (SdTz1):  Miramar NAS & Lindberg Fld.
>>** ;
 SdTz2Mn = (1*S93107Mn + 2*S42706Mn) / (1+2) ;  **<< Inland  (SdTz2):  Miramar NAS & El
Cajon(Santee) >>** ;

 SdTz1Av = (1*S93107Av + 2*S47740Av) / (1+2) ;  **<< Coastal (SdTz1):  Miramar NAS & Lindberg Fld.
>>** ;
 SdTz2Av = (1*S93107Av + 2*S42706Av) / (1+2) ;  **<< Inland  (SdTz2):  Miramar NAS & El Cajon(Santee)
>>** ;

 Avg =    ((Coastal) * SdTz1Av + (Inland) * SdTz2Av) / (Coastal + Inland) ;   **<< Do not round! >>** ;
 Max =    ((Coastal) * SdTz1Mx + (Inland) * SdTz2Mx) / (Coastal + Inland) ;   **<< Do not round! >>** ;
 Min =    ((Coastal) * SdTz1Mn + (Inland) * SdTz2Mn) / (Coastal + Inland) ;   **<< Do not round! >>** ;

 do over a_hd ;  ***<<< Round Degree-Days for each station. >>>*** ;
   a_hd = round(a_hd) ;
```

```
   a_cd = round(a_cd) ;
 end ;

 SdTz1Hd = (1*S93107Hd + 2*S47740Hd) / (1+2) ;  **<< Coastal (SdTz1):  Miramar NAS & Lindberg Fld.
>>** ;
 SdTz2Hd = (1*S93107Hd + 2*S42706Hd) / (1+2) ;  **<< Inland  (SdTz2):  Miramar NAS & El Cajon(Santee)
>>** ;

 SdTz1Cd = (1*S93107Cd + 2*S47740Cd) / (1+2) ;  **<< Coastal (SdTz1):  Miramar NAS & Lindberg Fld.
>>** ;
 SdTz2Cd = (1*S93107Cd + 2*S42706Cd) / (1+2) ;  **<< Inland  (SdTz2):  Miramar NAS & El Cajon(Santee)
>>** ;

 Hdd =    ((Coastal) * SdTz1Hd + (Inland) * SdTz2Hd) / (Coastal + Inland) ;    **<< Do not round! >>** ;
 Cdd =    ((Coastal) * SdTz1Cd + (Inland) * SdTz2Cd) / (Coastal + Inland) ;    **<< Do not round! >>** ;


 drop Coastal Inland ;

run ;




proc sort data=DailySD ;
   by year month ;
run ;

data LbDly.DailySD ;  **<< Save a copy of ALL the Daily Temperature Variables by Station and Sys-Avg for
later use. >>** ;
   set DailySD ;
run ;


data LbDly.SavgSDGE ;  **<< Save  JUST Daily System Avgerage Temperature Variables for later use.
>>** ;
   set DailySD ;
  format date date9. ;
  keep date  avg max min hdd cdd ;
run ;



proc means data=DailySD nway noprint;
     class year month;
     var hdd
         S42706Hd   S47740Hd    S93107Hd
         cdd
         S42706Cd   S47740Cd    S93107Cd  ;
     output out=MonSDGE sum=;
title3 "(Heating Degree-Days by Calendar Month)" ;
run ;



data MonSDGE ;
     set MonSDGE ;   *** Used to be named "MonSD.sd2" on 12/07/2004. *** ;

   date = mdy(month,1,year) ;
*** NOTE: Previously, (12/07/2004), exported as ".dbf" file a copy of "MonthSD.sd2" under the name
"MonthSCG.dbf" ... *** ;
```

```
*** Subsequent UPDATES using the SoCalGas (SCG) system-avg-calc method are to "MonSDGE.dbf" ...
*** ;
run;



/********************
proc dbload dbms=dbase data=MonSDGE ;   **<< Export a ".dbf" version of "MonSDGE.sd2" for later use.
>>** ;
                                   **<< Note:  Need to DELETE the previously-created ".dbf" file! >>** ;
 ** path='S:\Weather\2016Tcap-Phase II\SDGandE-Alt2Wgt\MonSDGE.dbf';
  path='C:\Weather\2016Tcap-Phase II\SDGandE-Alt2Wgt\MonSDGE.dbf';
  limit=0;
  load;
run;
********************/



proc sort data=MonSDGE  out=monthly(keep=date year month hdd cdd);
     by year month;
run;

* For saving non-rounded monthly numbers;
data lb2.MonSDGE;
            set MonSDGE;
run;



/***********
***********/
data lb2.mn72_21 ;  **<< Save a copy of the System Average Monthly Hdd/Cdd data for later use. >>** ;
   set monthly ;
  keep date year month hdd cdd ;
run ;




data ByStnDD ;
   set MonSDGE ;

  dateYYMM = mdy(month,1,year) ;

  if (dateYYMM < mdy(01,01,2022)) ;  **<< Only Pre January-2022 observations! >>** ;

  hdd = round(hdd) ;  **<< Round NOW for Monthly Reporting! >>** ;
  cdd = round(cdd) ;  **<< Round NOW for Monthly Reporting! >>** ;


   label S42706Hd  ='El Cajon - Hdd';
   label S42706Cd  ='El Cajon - Cdd';
   label S93107Hd  ='Miramar NAS - Hdd';
   label S93107Cd  ='Miramar NAS - Cdd';
   label S47740Hd  ='S.D. (Lindberg Fld.) - Hdd';
   label S47740Cd  ='S.D. (Lindberg Fld.) - Cdd';

   label hdd = "Syst-Avg. Hdd" ;
   label cdd = "Syst-Avg. Cdd" ;

run ;
```

```
proc sort data=ByStnDD ;
   by year month ;
run ;

data lb2.ByStnDD ;  **<< Save a copy of the MONTHLY Hdd/Cdd by Station for later use. >>** ;
   set ByStnDD ;
run ;







proc print data=ByStnDD  uniform split="/" ;
 id year;
 by year;
 sumby year;
     var dateYYMM hdd
         S42706Hd   S47740Hd    S93107Hd ;
     sum hdd
         S42706Hd   S47740Hd    S93107Hd ;
     format  dateYYMM worddate3.  hdd
         S42706Hd   S47740Hd    S93107Hd   6. ;

  label s42706Hd = "El Cajon   /              / ----------"
        s93107Hd = "Miramar    / Naval Air St./ ----------"
        s47740Hd = "San Diego  / Lindberg Fld./ ----------"

        hdd      = "System Avg./ for SDGandE  / ----------" ;


    ***<<< Note: For S# for El Cajon is 42706=CoopId# from NWS Weather Station Id# list, and
>>>*** ;
    ***<<<       the S# for San Diego, Lindberg Field is 47740=CoopId# from NWS Weather Station Id# list.
>>>*** ;
    ***<<<    But, the S# for Miramar, Naval Air Station is 93107=WBanID# from NWS Weather Station Id#
list. >>** ;

title3 "(Heating Degree-Days by Calendar Month)" ;

run ;



proc means data=ByStnDD noprint ;
   class year ;
     var hdd
         S42706Hd   S47740Hd    S93107Hd ;
     output out=HdSmByYr sum=;
run ;

/**
data HdSmByYr ;
  set HdSmByYr ;
  if (year=.) then delete ;
```

```
run ;
**/

proc print data=HdSmByYr  uniform split="/" ;
  where year ne . ;
     var year hdd
         S42706Hd   S47740Hd    S93107Hd ;
     format   hdd
         S42706Hd   S47740Hd    S93107Hd   6. ;

  label s42706Hd = "El Cajon   /              / ----------"
        s93107Hd = "Miramar    / Naval Air St./ ----------"
        s47740Hd = "San Diego  / Lindberg Fld./ ----------"

        hdd      = "System Avg./ for SDGandE  / ----------" ;


    ***<<< Note: For S# for El Cajon is 42706=CoopId# from NWS Weather Station Id# list, and
>>>*** ;
    ***<<<       the S# for San Diego, Lindberg Field is 47740=CoopId# from NWS Weather Station Id# list.
>>>*** ;
    ***<<<    But, the S# for Miramar, Naval Air Station is 93107=WBanID# from NWS Weather Station Id#
list. >>** ;

title3 "(Heating Degree-Days by Calendar Month)" ;

run ;
```

```
proc print data=ByStnDD  uniform split="/" ;
 id year;
 by year;
 sumby year;
     var dateYYMM Cdd
         S42706Cd   S47740Cd    S93107Cd ;
     sum Cdd
         S42706Cd   S47740Cd    S93107Cd ;
     format  dateYYMM worddate3.  Cdd
         S42706Cd   S47740Cd    S93107Cd   6. ;

  label s42706Cd = "El Cajon   /              / ----------"
        s93107Cd = "Miramar    / Naval Air St./ ----------"
        s47740Cd = "San Diego  / Lindberg Fld./ ----------"

        Cdd      = "System Avg./ for SDGandE  / ----------" ;


    ***<<< Note: For S# for El Cajon is 42706=CoopId# from NWS Weather Station Id# list, and
>>>*** ;
    ***<<<       the S# for San Diego, Lindberg Field is 47740=CoopId# from NWS Weather Station Id# list.
>>>*** ;
    ***<<<    But, the S# for Miramar, Naval Air Station is 93107=WBanID# from NWS Weather Station Id#
list. >>** ;
```

```
title3 "(Cooling Degree-Days by Calendar Month)" ;

run ;



proc means data=ByStnDD noprint ;
   class year ;
     var Cdd
         S42706Cd   S47740Cd    S93107Cd ;
     output out=CdSmByYr sum=;
run ;

/**
data CdSmByYr ;
  set CdSmByYr ;
  if (year=.) then delete ;
run ;
**/

proc print data=CdSmByYr  uniform split="/" ;
  where year ne . ;
     var year Cdd
         S42706Cd   S47740Cd    S93107Cd ;
     format   Cdd
         S42706Cd   S47740Cd    S93107Cd   6. ;

  label s42706Cd = "El Cajon   /              / ----------"
        s93107Cd = "Miramar    / Naval Air St./ ----------"
        s47740Cd = "San Diego  / Lindberg Fld./ ----------"

        Cdd      = "System Avg./ for SDGandE  / ----------" ;


    ***<<< Note: For S# for El Cajon is 42706=CoopId# from NWS Weather Station Id# list, and
>>>*** ;
    ***<<<       the S# for San Diego, Lindberg Field is 47740=CoopId# from NWS Weather Station Id# list.
>>>*** ;
    ***<<<    But, the S# for Miramar, Naval Air Station is 93107=WBanID# from NWS Weather Station Id#
list. >>** ;

title3 "(Cooling Degree-Days by Calendar Month)" ;

run ;







***<< Print Summary Tables of Hdd and Cdd by month  with Annual Totals >>*** ;

proc transpose data=monthly out=hddSmry prefix=mon ;
   by year;
   id month ;
   var hdd ;
run ;
```

```
data hddSmry ;
   set hddSmry ;
 hddTot = sum(of mon1-mon12) ;
run ;

proc print data=hddSmry ;
 id year ;
 var hddTot mon1-mon12 ;
title1 'Monthly Heating Degree-Days from 1972 thru 2021(Month-to-Date)';
title2 "   " ;
run ;



proc transpose data=monthly out=cddSmry prefix=mon ;
   by year;
   id month ;
   var cdd ;
run ;

data cddSmry ;
   set cddSmry ;
 cddTot = sum(of mon1-mon12) ;
run ;

proc print data=cddSmry ;
 id year ;
 var cddTot mon1-mon12 ;
title1 'Monthly Cooling Degree-Days from 1972 thru 2021(Month-to-Date)';
run ;







quit ;
```

SDG&E SAS Code #2: MinAvg-Freq(SDGandE)ByMonth.sas

```
title1 "Calculate Min{Avg} (Minimum Average Daily Temp.) by Months for all data over a specified range of
YEARS." ;

/*****************************************************************************************************

   File: S:\Weather\2016 CGR\SDGE123!@#qwe1
\MinAvg-Freq(SDGandE)ByMonth.sas

 *****************************************************************************************************/


options date number source notes ;
options mprint ;

/* %cour8l
  %cour8p; */

 options ls=160 ps=90 ;  *<To get PORTRAIT  and SAS-Monospace w/Roman 6pt. FONT >*;
*options ls=211 ps=69 ;  *<To get LANDSCAPE and SAS-Monospace w/Roman 6pt. FONT >*;

**options nomprint ;


    **<< Data set "SysAvgSD.sd2" was created by SAS program on file:                    >>** ;
    **<<     "S:\Weather\2016Tcap-Phase II\SDGandE-Alt2Wgt\SysAvTmp(SCG-Method)_CustCntWgts-
v2.sas"      >>** ;
    **<<                                                                     >>** ;
libname in '/EDS_RB/Weather/2022 CGR/SDGE';



  **<< Directory to save a copy of output data set! >>** ;
libname out '/EDS_RB/Weather/2022 CGR/SDGE';


proc contents data=in.SAvgSDGE ;
run ;


     *** << Note: Need to change here and below!  Don't forget! >> *** ;
%let startyr=1972;  ***<< Value of "Start Year" >>***;
%let lastyr=2021 ;  ***<< Value of "Last Year" >>***;

/********
%let tgtmonth=xx ;  ***<< Value of "Target Month", i.e., 1,2,3, ... 12 >>** ;
********/




proc format ;
  value mmmmmm    1='Jan'
             2='Feb'
             3='Mar'
             4='Apr'
             5='May'
             6='Jun'
             7='Jul'
             8='Aug'
             9='Sep'
```

```
                10='Oct'
                11='Nov'
                12='Dec'
                13='Min4Yr' ;
run ;


%macro FreqMon(name_mon,tgtmonth) ;

data combined ;
    set in.SAvgSDGE ;

year = year(date) ;
month= month(date) ;
day = day(date) ;
  **<< To "Select" Winter Months Only. >>**;
  **  if month in (1,2,3,11,12);

*** hdd = round(avg,1);  **<< Comment out ... Do NOT round data this time! >>** ;
if ((year >= &startyr) & (year <= &lastyr)) ;

%if (&tgtmonth >= 1 and &tgtmonth <= 12) %then
  %do ;
if (month = &tgtmonth) ;  **<< To select only a specific month! >>** ;
  %end ;
%else
  %do ;
month = 13 ;  **<< Set "month" variable to "13" and select ALL "months" of the YEAR! >>** ;
  %end ;
run;


proc sort data=combined;
 by year month day ;
run;

***proc contents data=combined ;
***run ;

proc means data=combined noprint;
  by year month ;
  var avg ;
  output out=&name_mon min=MinAvg;
  title1 "Minium{avg} (Minimum Avg. Daily Temp.) for &name_mon  by YEAR=&startyr to &lastyr";
run;

proc print data=&name_mon ;
run ;

%mend ;








%let startyr=1972;  ***<< Value of "Start Year" >>***;
%let lastyr=2021 ;  ***<< Value of "Last Year" >>***;
```

```
/************
%let name_mon='January ';  ***<< NAME of "Target Month" >>***;
%let tgtmonth= 1 ;     ***<< Value of "Target Month" >>***;
************/

%FreqMon(JAN,1) ;

data AllMonth ;
 set JAN ;
run ;


%FreqMon(FEB,2) ;

data AllMonth ;
 set AllMonth FEB ;
run ;


%FreqMon(MAR,3) ;

data AllMonth ;
 set AllMonth MAR ;
run ;


%FreqMon(APR,4) ;

data AllMonth ;
 set AllMonth APR ;
run ;


%FreqMon(MAY,5) ;

data AllMonth ;
 set AllMonth MAY ;
run ;


%FreqMon(JUN,6) ;

data AllMonth ;
 set AllMonth JUN ;
run ;


%FreqMon(JUL,7) ;

data AllMonth ;
 set AllMonth JUL ;
run ;


%FreqMon(AUG,8) ;

data AllMonth ;
 set AllMonth AUG ;
run ;
```

```
%FreqMon(SEP,9) ;

data AllMonth ;
  set AllMonth SEP ;
run ;


%FreqMon(OCT,10) ;

data AllMonth ;
  set AllMonth OCT ;
run ;


%FreqMon(NOV,11) ;

data AllMonth ;
  set AllMonth NOV ;
run ;


%FreqMon(DEC,12) ;

data AllMonth ;
  set AllMonth DEC ;
run ;


%FreqMon(ALL,13) ;

data AllMonth ;
  set AllMonth All ;
run ;



proc Tabulate data=AllMonth;
  class year month ;
  var MinAvg ;
  table year, MinAvg*f=6.2*(month)/rts=6 ;
  label MinAvg='Min{Avg} by Mo' ;
 *keylabel All ='Min4Yr';
  format month mmmmmm. ;
  title2 "Min{avg} by Months for all Years from YEAR=&startyr to &lastyr";
run;


proc sort data=AllMonth out=MinAvg_d;
  by month year MinAvg ;
run ;


proc print data=MinAvg_d ;
  by month ;
  pageby month ;
  var year MinAvg ;
run ;


data out.MinAvg_d ;  *<< Save a copy for later use!  >>** ;
  set MinAvg_d ;
```

```
 drop _freq_ _type_ ;
run ;

proc contents data=out.MinAvg_d ;
run ;


/************************
 **<< Export a copy as ".dbf" as well. >>** ;
 **<< Note: Must "delete" the prior version, othewise the save will not execute. >>** ;
proc dbload dbms=dbase data=MinAvg_d ;
path='/data/home/scgdf/2016_CGR_Weather/SDGE/MinAvg_d.dbf';
limit=0;
load;
run;
*************************/



quit ;
```

SDG&E SAS Code #3: GEV4DlyTemp(NLReg2)ByMonthMACRO_Sdge.sas

```
Title1 "Data Analysis for Maximum/Minimum Daily SysAvg Temperatures (Un-Rounded)." ;
Title2 "Fit GEV Probability Model to Empirical CDF using NL-OLS Regression Methods." ;

/**************************************************************************************************************/
/*                                                                                    */
/*                                                                                    */
/*                                                                                    */
/* FILE SAVED: "S:\Weather\2016 Cgr\SDGE\GEV4DlyTemp(NLReg2)ByMonthMACRO_Sdge.sas"
*/
/*          for Annual Max of Negative of Min. Temp.                                        */
/*          for each of calendar months Jan-Dec, and the entier year(index=13).                   */
/*                                                                                    */
/*          Fit GEV models (3-parameter and 2-parameter), plus a simple T-Distribution model.        */
/*                                                                                    */
/*                                                                                    */
/*************************************************************************************************&***************/



options mprint ;
/* %cour8p
  %cour8l */


 options ls=211 ps=69 ; **<<LANDSCAPE:  SAS-Monospace w/Roman 6pt. Font >>**;
*options ls=160 ps=90 ; **<<PORTRAIT:   SAS-Monospace w/Roman 6pt. Font >>**;

options date number notes ;



libname out1 '/EDS_RB/Weather/2022 CGR/SDGE' ;  **<< Directory for daily weather variables as INPUT.
>>**;

libname out2 '/EDS_RB/Weather/2022 CGR/SDGE/MinTemp' ;  **<< Directory for estimation results OUPUT
files. >>** ;



proc contents data=out1.SAvgSDGE ;
run ;

data seriesD ;
  set out1.SAvgSDGE ;
  year = year(date) ;
  month = month(date) ;
  posAvg = avg ;
  negAvg = -avg ;
run ;


proc means data=seriesD noprint nway ;
 class year month ;
  var posAvg negAvg ;
 output out=mostat
                mean=posAvg  negAvg
                 max=MxPosAvg MxNegAvg
                 min=MnPosAvg MnNegAvg    ;
run;
```

```
proc sort data=mostat ;
   by year month ;
run ;


data mostat ;
   set mostat ;
   MxPRatio = MxPosAvg/ PosAvg  ;
   MnPRatio = MnPosAvg/ PosAvg  ;
   MxNRatio = MxNegAvg/ NegAvg  ;
   MnNRatio = MnNegAvg/ NegAvg  ;
run ;







/****************
***<< Print Summary Tables of Means/Minimums/Maximums of daily NEGATIVE-Temperatures (degrees-F).
>>*** ;

proc transpose data=mostat out=AvTData prefix=AvT_ ;
   where (year < 2016) ;
   by year;
   id month ;
   var NegAvg  ;
run ;

data AvTData ;
   set AvTData ;

 if (mod(year,4)=0) then do ;
        AvT_13 = (AvT_1 + AvT_3 + AvT_5 + AvT_7 + AvT_8 + AvT_10 + AvT_12)*31
               + (AvT_4 + AvT_6 + AvT_9 + AvT_11)*30
               + (AvT_2)*29 ;
        AvT_13 = AvT_13 / 366 ;
        end ;
 else do ;
        AvT_13 = (AvT_1 + AvT_3 + AvT_5 + AvT_7 + AvT_8 + AvT_10 + AvT_12)*31
               + (AvT_4 + AvT_6 + AvT_9 + AvT_11)*30
               + (AvT_2)*28 ;
        AvT_13 = AvT_13 / 365 ;
        end ;

run ;

proc print data=AvTData ;
 id year ;
 var AvT_13 AvT_1-AvT_12 ;
title3 'Monthly Mean NEGATIVE Temperature (Deg-F) from 1972 thru 2015.';
run ;





proc transpose data=mostat out=MnTData prefix=MnT_ ;
   where (year < 2016) ;
```

```
   by year;
   id month ;
   var MnNegAvg ;
run ;

data MnTData ;
   set MnTData ;
 MnT_13 = min(of MnT_1-MnT_12) ;
run ;

proc print data=MnTData ;
 id year ;
 var MnT_13 MnT_1-MnT_12 ;
title3 'Monthly MINIMUM NEGATIVE-Temperature (Deg-F) from 1972 thru 2015.';
run ;
****************/




proc transpose data=mostat out=MxTData prefix=MxT_ ;
   where (year < 2022) ;
   by year;
   id month ;
   var MxNegAvg ;
run ;

data MxTData ;
   set MxTData ;
 MxT_13 = max(of MxT_1-MxT_12) ;
run ;

proc print data=MxTData ;
 id year ;
 var MxT_13 MxT_1-MxT_12 ;
title3 'Monthly MAXIMUM NEGATIVE-Temperature (Deg-F) from 1972 thru 2021.';
run ;










/****************
***<< Descriptive Statistics: Maximums of daily NEGATIVE-Temperatures (Deg-F) for Year and each
calendar month. >>*** ;


proc corr data=MxTData ;
  var MxTyr MxT_1 - MxT_12 ;
title3 'Correlation Matrix of Monthly Maximum NEGATIVE-Temperatures (Deg-F) within same year.';
run ;

proc arima data=MxTData ;
  identify var=MxT_13 ;
  identify var=MxT_1 ;
  identify var=MxT_2 ;
```

```
  identify var=MxT_3 ;
  identify var=MxT_4 ;
  identify var=MxT_5 ;
  identify var=MxT_6 ;
  identify var=MxT_7 ;
  identify var=MxT_8 ;
  identify var=MxT_9 ;
  identify var=MxT_10 ;
  identify var=MxT_11 ;
  identify var=MxT_12 ;
title3 "Auto-correlation analysis of each calendar month's Maximum NEGATIVE-Temperatures (Deg-F)
within same year.";
run ;

proc univariate normal data=MxTData plot ;
  id year ;
  var MxT_13 MxT_1 - MxT_12 ;
title3 "Probability plots and tests for NORMALity by each calendar month's Maximun NEGATIVE-
Temperatures (Deg-F) time series.";
run ;


proc means data=MxTData ;
  var MxT_1 - MxT_12  MxT_13 ;
run ;
****************/
```

```
***<< Statistical Estimation of GEV Models: Maximums of daily heating degrees for Year and each calendar
month. >>*** ;

%macro RankIt(file=MxTData,var=MxT_13,rank=Rank,prob=PrMxT_13,Nobser=50,PltValue=0.375) ;
proc sort data=&file ;
  by &var ;
run ;

data &file ;
   set &file ;
   retain &rank 0     alpha &pltvalue ;

  &rank = &rank + 1 ;
  &prob =  (&rank - alpha) / (&Nobser +(1 - 2*alpha)) ;
run ;

proc print data=&file ;
  var &var &rank &prob alpha year ;
run ;
%mend RankIt ;



%macro
GEVfit(file=MxTData,ofile=MxTNL1,outfit=fit1,outest=est1,depvar=PrMxT_13,var=MxT_13,typeGEV=1,
             KappaI=0.25,GammaI=-47.05,ThetaI=2.77,YrLo=1972,YrHi=2021) ;
proc sort data=&file ;
```

```
   by year ;
run ;



proc model data=&file converge=0.001
          maxit=500 dw ;  outmodel=&ofile ;
     range year = &YrLo to &YrHi ;  **<< Dropped Jan-Feb 2016 data. >>** ;


     y = (&var - Gamma) / Theta ;

    %if &typeGEV=1 %then %do ;  ***<< 3-parameter GEV Model. >>>*** ;
        &depvar = exp( -(1 - Kappa * (y))**(1/Kappa) ) ;
        %let typmod = 3-parameter GEV Model. ;
        %end ;

    %if &typeGEV=2 %then %do ;  **<< 2-parameter "Double Exponential" or  "Gumbel" Model. >>** ;
        &depvar = exp( -exp(-(y)) ) ;
        %let typmod = 2-parameter Double Exponential or Gumbel Model. ;
        %end ;

    %if (&typeGEV NE 1) AND (&typeGEV NE 2) %then  %do ; **<< 2-parameter "T-Dist" Model. >>** ;
        dft=(&YrHi - &YrLo) +1 -2 ;
        &depvar = probt(y,dft) ;
        %let typmod = 2-parameter T-Dist Model. ;
        %end ;


  %if &typeGEV = 1 %then %do ;
  parms
        Kappa  &KappaI
        Gamma  &GammaI
        Theta  &ThetaI ;
  %end ;

  %if (&typeGEV NE 1)  %then %do ;
  parms
        Gamma  &GammaI
        Theta  &ThetaI ;
  %end ;


  fit &depvar   /out=&outfit  outall
              outest=&outest corrb corrs outcov ;

title3 "Non-linear Estimation of &&typmod: for Maximum NEGATIVE Temperature (Deg-F)." ;
run ;
%mend GEVfit ;







%macro GEVbyMo(mm=_1) ;


*********<<< Analysis for "January"    (i.e., SUFIX "mm" = "_1"  >>>******************;
*********<<< Analysis for "February"   (i.e., SUFIX "mm" = "_2"  >>>******************;
*********<<< Analysis for "March"      (i.e., SUFIX "mm" = "_3"  >>>******************;
*********<<< Analysis for "April"      (i.e., SUFIX "mm" = "_4"  >>>******************;
```

```
*********<<< Analysis for "May"       (i.e., SUFIX "mm" = "_5"  >>>******************;
*********<<< Analysis for "June"      (i.e., SUFIX "mm" = "_6"  >>>******************;
*********<<< Analysis for "July"      (i.e., SUFIX "mm" = "_7"  >>>******************;
*********<<< Analysis for "August"    (i.e., SUFIX "mm" = "_8"  >>>******************;
*********<<< Analysis for "September" (i.e., SUFIX "mm" = "_9"  >>>******************;
*********<<< Analysis for "October"   (i.e., SUFIX "mm" = "_10"  >>>******************;
*********<<< Analysis for "November"  (i.e., SUFIX "mm" = "_11"  >>>******************;
*********<<< Analysis for "December"  (i.e., SUFIX "mm" = "_12"  >>>******************;
*********<<< Analysis for "ALL Months"  (i.e., SUFIX "mm" = "_13"  >>>******************;

%RankIt(file=MxTData,var=MxT&mm,rank=Rank&mm,prob=PrMxT&mm,Nobser=50,PltValue=0.375) ;




%GEVfit(file=MxTData,ofile=MxTNL2,outfit=fit2,outest=est2,depvar=PrMxT&mm,var=MxT&mm,typeGEV=2,
       KappaI=0.25,GammaI=&&gamma&mm,ThetaI=&&theta&mm,YrLo=1972,YrHi=2021) ;


proc print data=fit2 ;
run ;




proc transpose data=fit2 out=pred2 prefix=probP_ ;
   where (_type_ = "PREDICT" ) ;
   by year;
   var prmxt&mm ;
run ;

data comb2 ;
   merge MxTData pred2 ;
  by year ;
   ProbP2 = ProbP_1 ;
  keep year MxT&mm PrMxT&mm ProbP2 ;
run ;


proc print data=comb2 ;
run ;


proc plot data=comb2 ;
  plot prmxt&mm*MxT&mm='*'
      probP2*MxT&mm='-' / overlay ;
run ;


proc print data=est2 ;
run ;



data out2.est2&mm ;   ***<<< Save a copy of the "Double Exponential Model" estimation results!  >>>*** ;
  set est2 ;
run ;
```

```
%GEVfit(file=MxTData,ofile=MxTNL0,outfit=fit0,outest=est0,depvar=PrMxT&mm,var=MxT&mm,typeGEV=0,
        KappaI=0.25,GammaI=&&gamma&mm,ThetaI=&&theta&mm,YrLo=1972,YrHi=2021) ;

proc print data=fit0 ;
run ;

proc transpose data=fit0 out=pred0 prefix=probP_ ;
   where (_type_ = "PREDICT" ) ;
   by year;
   var prmxt&mm ;
run ;

data comb0 ;
   merge MxTData pred0 ;
  by year ;
   ProbP0 = ProbP_1 ;
  keep year MxT&mm PrMxT&mm ProbP0 ;
run ;

proc print data=comb0 ;
run ;

proc plot data=comb0 ;
  plot prmxt&mm*MxT&mm='*'
       probP0*MxT&mm='-' / overlay ;
run ;

proc print data=est0 ;
run ;

data out2.est0&mm ;   ***<<< Save a copy of the 2-parameter "T-Distribution" Model estimation results!
>>>*** ;
  set est0 ;
run ;

%GEVfit(file=MxTData,ofile=MxTNL1,outfit=fit1,outest=est1,depvar=PrMxT&mm,var=MxT&mm,typeGEV=1,
        KappaI=0.25,GammaI=&&gamma&mm,ThetaI=&&theta&mm,YrLo=1972,YrHi=2021) ;

proc print data=fit1 ;
run ;

proc transpose data=fit1 out=pred1 prefix=probP_ ;
   where (_type_ = "PREDICT" ) ;
   by year;
   var prmxt&mm ;
run ;
```

```
data comb1 ;
   merge MxTData pred1 ;
  by year ;
   ProbP1 = ProbP_1 ;
  keep year MxT&mm PrMxT&mm ProbP1 ;
run ;


proc print data=comb1 ;
run ;


proc plot data=comb1 ;
  plot prmxt&mm*MxT&mm='*'
       ProbP1*MxT&mm='-' / overlay ;
run ;



proc print data=est1 ;
run ;



data out2.est1&mm ;   ***<<< Save a copy of the "G.E.V. Model" estimation results!  >>>*** ;
  set est1 ;
run ;

%mend GEVbyMo ;

/****************
****************/

proc means data=MxTData ;
  var MxT_1 - MxT_12  MxT_13;
 output out=VarStat
        mean=mean1-mean12 mean13
        std=stdev1-stdev12 stdev13;
title3 "Calc. Means and Standard Deviantions to use as Starting Values in Non-Linear Estimations." ;
run ;


proc print data=VarStat ;
run ;

data _null_ ;
   set VarStat ;

  call symput('gamma_13',mean13) ;
  call symput('theta_13',stdev13) ;

  call symput('gamma_12',mean12) ;
  call symput('theta_12',stdev12) ;

  call symput('gamma_11',mean11) ;
  call symput('theta_11',stdev11) ;

  call symput('gamma_10',mean10) ;
  call symput('theta_10',stdev10) ;

  call symput('gamma_9',mean9) ;
```

```
  call symput('theta_9',stdev9) ;

  call symput('gamma_8',mean8) ;
  call symput('theta_8',stdev8) ;

  call symput('gamma_7',mean7) ;
  call symput('theta_7',stdev7) ;

  call symput('gamma_6',mean6) ;
  call symput('theta_6',stdev6) ;

  call symput('gamma_5',mean5) ;
  call symput('theta_5',stdev5) ;

  call symput('gamma_4',mean4) ;
  call symput('theta_4',stdev4) ;

  call symput('gamma_3',mean3) ;
  call symput('theta_3',stdev3) ;

  call symput('gamma_2',mean2) ;
  call symput('theta_2',stdev2) ;

  call symput('gamma_1',mean1) ;
  call symput('theta_1',stdev1) ;

run ;


%GEVbyMo(mm=_13) ;   **<< Annual(Entire Year of Data.) >>** ;


%GEVbyMo(mm=_1) ;   **<< Jan Data. >>** ;


%GEVbyMo(mm=_2) ;   **<< Feb Data. >>** ;
%GEVbyMo(mm=_3) ;   **<< Mar Data. >>** ;


%GEVbyMo(mm=_4) ;   **<< Apr Data. >>** ;


%GEVbyMo(mm=_5) ;   **<< May Data. >>** ;
%GEVbyMo(mm=_6) ;   **<< Jun Data. >>** ;
%GEVbyMo(mm=_7) ;   **<< Jul Data. >>** ;


%GEVbyMo(mm=_8) ;   **<< Aug Data. >>** ;


%GEVbyMo(mm=_9) ;   **<< Sep Data. >>** ;
%GEVbyMo(mm=_10) ;   **<< Oct Data. >>** ;
%GEVbyMo(mm=_11) ;   **<< Nov Data. >>** ;
%GEVbyMo(mm=_12) ;   **<< Dec Data. >>** ;

quit ;
```