Automated Unimpaired Hydrologic Metric Scaling for California Streams

Abstract

The following report is the beginning of a larger study to provide unimpaired daily hydrographs for streams in the state of California. The larger project objective is to create an ArcGIS based tool that will automatically produce streamflow estimates for any reach in California. This future method will use reference gauges and to-be-determined scaling methods. This paper reviews the available scaling methods and compares them for accuracy and dependability. Data was obtained from ArcGIS Online and Dr. Belize Lane at Utah State University. ArcGIS was used to organize, calculate, and select data. A large portion of the research involved coding scenarios in Visual Basic and Excel to test different scalar methods. Results of the project are preliminary but show that traditional scaling approaches can provide accurate predictions with some limitations. They also show that a classification system for providing metrics may be better at predicting hydrograph shapes than traditional methods alone.

By

Karl Christensen

GIS Water Resources CEE 6440

December 2017

Introduction

The objective of this research project is to provide unimpaired flow metrics for any stream reach in California. These reference flow metrics are characteristics of the timing, magnitude, duration, frequency and rate of change (Poff et al, 1997) of daily streamflow time series in the absence of major alterations by dams, diversions, and land use changes. Reference flow metrics have been linked to ecological integrity and can be used to guide flow management decisions to restore or retain ecological objectives. Predicting reference flow metrics in stream reaches without local gauge data is an existing challenge. Here, several alternative approaches have been evaluated for scaling streamflow data from existing reference gauges to ungauged locations and their ability to predict a set of reference flow metrics is compared.

An unimpaired daily flow regime directly affects stream ecology. Hydrology and aquatic biodiversity have been linked via four key mechanisms: a) flow is a major determinant of the habitat, a key driver of the aquatic composition, b) aquatic species have evolved life-history strategies in response to the natural flow regime, c) the natural pattern of the longitudinal and lateral connectivity in the river system is important for supporting populations of aquatic species and d) the invasion and success of non-native species is facilitated by alterations to streamflow (Bunn and Arthington, 2002). Alteration of the natural flow regime often leads to ecological degradation and the shifting of species assemblage away from native species (Chinnayakanahalli, 2010).

Several traditional scaling methods are established and used to predict streamflow when a reference stream gauge is not available. Farmer and Vogel (Farmer and Vogel, 2012) list three of these methods:

1. Scaling flows by the Drainage-Area Ratio (DAR) technique
2. Scaling flows by average streamflow
3. Scaling flow by average and standard deviation of streamflow

The DAR scalar is a common scaling method because it only requires catchment areas of the reference and prediction sites and streamflow data from the reference site. Methods 2 and 3 require streamflow data estimates from both the reference and prediction sites but no drainage areas. While these methods are useful for direct interpolation, they do not account hydrologic variability in the streams being compared. Potential variability not accounted for in streamflow averages or catchment areas may negatively influence the results.

Stream flow variability can be better predicted when streams are compared across an entire region. Hersch and Maidment (2007) created a classification scheme that distinguished geographic regions of streams with similar attributes related to water quality, climatology, hydrology & hydraulics, geomorphology & physical processes, and biology. Five hydrologic regions were identified for the state of Texas: North-Central Texas, West Texas, East Texas, Lower Rio Grande Basin, and South-Central Texas (see Figure 1 below). Their results allowed further hydrologic analysis to recognize which streams will have similar behavior.

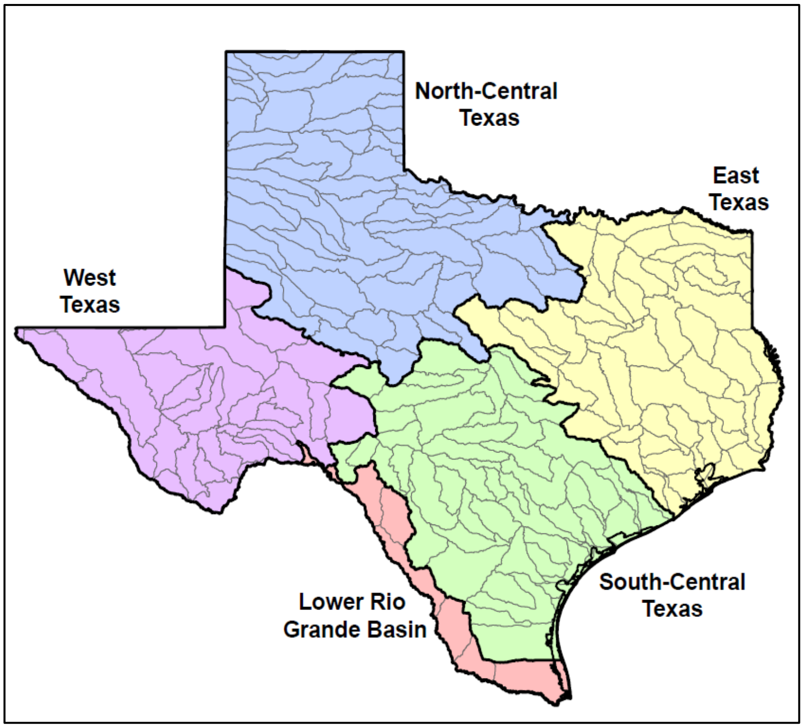

Figure 1 - Texas Classifications: Five Regions (Maidment and Hersch, 2007)

Similar to Maidment and Hersch, Lane et al (2017) distinguished nine hydrologic classes for the state of California comparing 20 different attributes. Unlike Maidment and Hersch, Lane et al did not attempt to build geographic boundaries but instead depended upon a statistical analysis of each individual stream's attributes (see Figure 2). The resulting stream classes can be seen in Figure 3 and show a much 'messier' regionalization with a more heterogeneous distribution. Each stream class is named according to the driving hydrologic conditions: Class 1 - Snowmelt, Class 2 - Low-volume snowmelt and rain, Class 3 - High-volume snowmelt and rain, Class 4 - Rain and seasonal groundwater, Class 5 - Winter storms, Class 6 - Groundwater, Class 7 - Perennial groundwater and rain, Class 8 - Flashy, ephemeral rain, and Class 9 - High elevation low precipitation. *Class 9 is not shown in Figure 3 because it was developed after the original publication of Lane et al 2017*.

Hydrologic Index
Mean annual flow
Annual C.V.
Flow predictability
% of floods in 60-day period
med_Oct
med_May
One-day minimum
Date of minimum
Date of maximum
Low pulse duration
High pulse count
Extreme low duration
Extreme low timing
High flow duration
High flow timing
Small flood duration
Small flood frequency
Large flood duration
Large flood timing
Large flood fall rate

Figure 2 – Hydrologic Indices (Lane et. al, 2017)

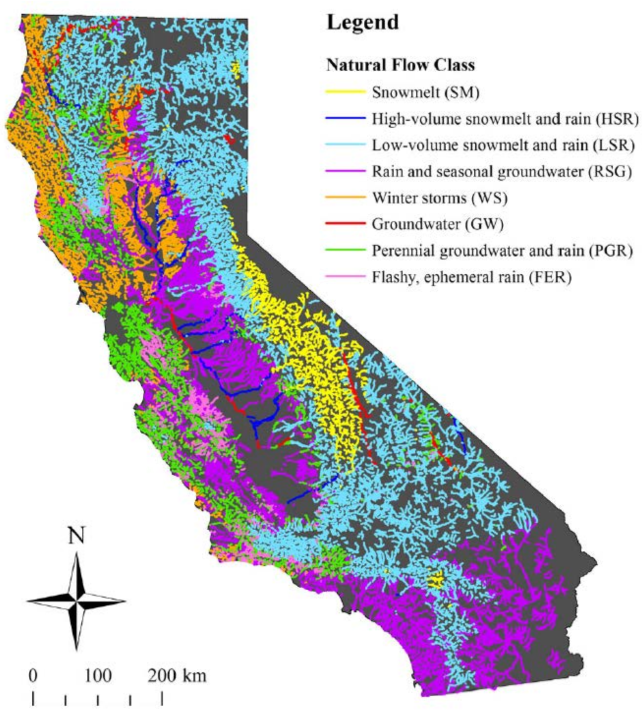

Figure 3 - California Stream Classes (Lane et. al, 2017)

Lane et al further defined the classes by creating Dimensionless Reference Hydrographs (DRH) for each stream and an average DRH for each class. The DRH values can be interpreted as metrics such as timing of yearly peak flows, duration of peak flows, lowest mean annual flows, and other hydrologic characteristics. As part of an assigned class, each DRH also holds certain physical and climatic catchment controls that allow most streams in California to be classified. Streams of the same classification can be assumed to hold similar DRH patterns and provide a foundation for developing alternative scaling methods.

Method of Work

Data from multiple sources were used to test each of the traditional and classification based scaling methods. Data provided by Dr. Belize Lane at Utah State University included the DRH values for individual gauge stations and class averages, actual stream flow data for gauges, and monthly & annual flow estimates for each predicted gauge. Drainage areas were calculated in ArcGIS Pro using the National Elevation Dataset at 30m resolution and the ArcGIS hydrology tool package. All data was combined first in ArcGIS Pro and then tabulated in Excel. The automation of calculating each method was accomplished using Visual Basic (VBA).

The traditional and classification scaling methods were expanded into 10 separate scenarios shown below in Table 1. Each scenario was added into the VBA code to predict a 20-year daily streamflow time series. The classification methods utilized DRH Values and Annual/Monthly Averages from the USGS (via Dr. Lane); the traditional methods utilized actual daily flows and calculated drainage areas depending on the scenario.

Table 1 - Scenarios Methods (Classification Method - Blue, Traditional Method - Green, Actual Daily Time Series - Red)

TYPE	SCENARIO	TIME SERIES	SCALARS
CLASSIFICATION	1	Aggregate DRH Values	Annual Averages
CLASSIFICATION	2	Aggregate DRH Values	Monthly Averages
CLASSIFICATION	3	Nearest 1 DRH Values	Annual Averages
CLASSIFICATION	4	Nearest 1 DRH Values	Monthly Averages
CLASSIFICATION	5	Nearest 3 DRH Values	Annual Averages
CLASSIFICATION	6	Nearest 3 DRH Values	Monthly Averages
TRADITIONAL	7	Nearest 1 Daily Flows	Drainage Area Ratio
TRADITIONAL	8	Nearest 1 Daily Flows	Annual Average Ratio
TRADITIONAL	9	Nearest 1 Daily Flows	Monthly Average Ratio
TRADITIONAL	10	Nearest 1 Daily Flows	Standard Deviation Ratio
NO SCALING	Actual	Prediction Site Daily Flows	N/A

Prediction and reference gauge sites were chosen in ArcGIS Pro by selecting an area with at least four gauge sites of the same class. Centermost sites were used as 'prediction sites' and gauges nearby as 'reference sites' (see Figure 4). USGS Gauge identification numbers, such as those shown in Figure 4, were then used as inputs in the Excel input worksheet. The VBA main code then ran all 10 scenarios for the 4 gauge sites and produced a 20-year daily time series for each scenario. *A copy of the VBA Code can be found in Appendix A.*

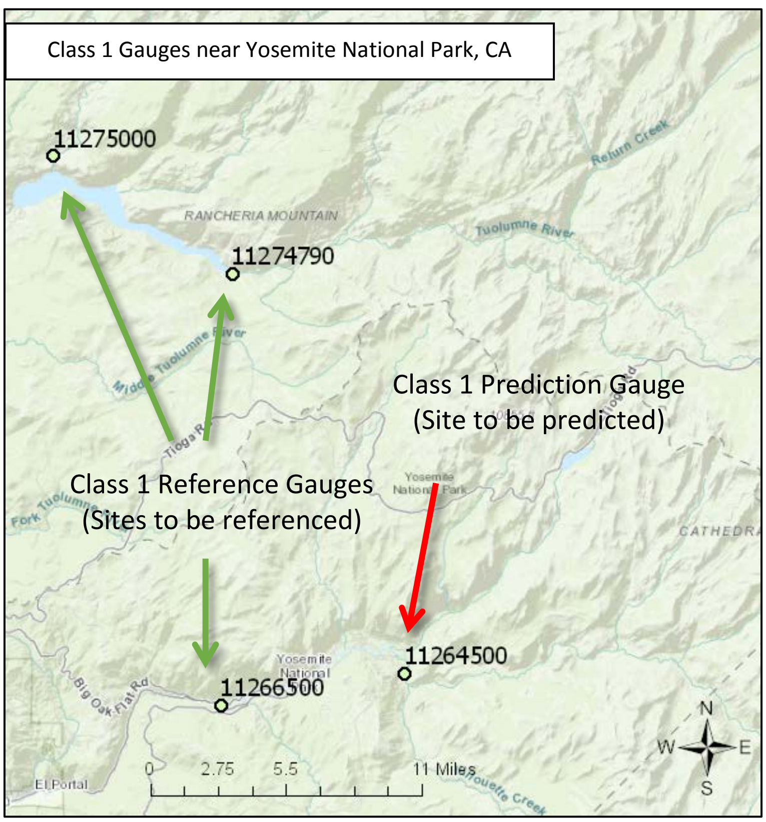

Figure 4 - Class 1 Gauges near Yosemite National Park

The hand calculations for building each time series and scalar can be seen in Figures 5 & 6. Notice that the time series are label A through D and the scalars are labeled 1 through 6. The scenarios then have labels such as A-1, B-2, D-4, and so on, depending on the combination of series and scalars. These calculations were used to confirm the accuracy of the program and spreadsheet values.

Hand Calcs for Scenario 1 - First Day in Calculation Shown Below 10/1/1968

Site Data=> Prediction Site: 11264500 | N1: 11266500 | N2: 11226500 | N3: 11275000

Time Series	Description	Source
A	Aggregate_DRH := 0.32320272	DRH Averages - Class 1
B	N1_DRH := 0.250324	DRH for 11266500
	N2_DRH := 0.279646	DRH for 11226500
	N3_DRH := 0.294403	DRH for 11275000
C	$NAverage\_DRH := \frac{N1\_DRH + N2\_DRH + N2\_DRH}{3} = 0.2699$	Calculation
D	N1_DailyFlow := 13 cfs	DRH for 11266500
Actual	PSite_DailyFlow := 2.6 cfs	Flow for 11264500

Scalar	Description	Source
1	Annual_Scalar := 673.71529 cfs	Site Data 11264500
2	Monthly_Scalar := 16.0635 cfs	Site Data 11264500
	PSite_Area := 468.31844999 km^2	Site Data 11266500
	N1_Area := 834.8886 km^2	Site Data 11266500
3	$DAR := \frac{PSite\_Area}{N1\_Area} = 0.5609$	Calculation
	PSite_AnnualAvg := 673.715 cfs	1989 WY Avg for 11264500
	N1_AnnualAvg := 1219.3452 cfs	1989 WY Avg for 11266500
4	$AnnualAvgRatio := \frac{PSite\_AnnualAvg}{N1\_AnnualAvg} = 0.5525$	Calculation
	PSite_MonthlyAvg := 48.326167 cfs	October Avg for 11264500
	N1_MonthlyAvg := 88.63161 cfs	October Avg for 11266500
5	$MonthlyAvgRatio := \frac{PSite\_MonthlyAvg}{N1\_MonthlyAvg} = 0.5452$	Calculation
6	Standard Deviation $\sigma_1 := 98.9580$ cfs	N1 October StandDev (σ)
	Standard Deviation $\sigma_2 := 47.7228$ cfs	PSite October StandDev (σ)
	Mean1 $\mu_1 := 88.6316$ cfs	N1 October Avg
	Mean2 $\mu_2 := 48.3262$ cfs	PSite October Avg

Figure 5 - Hand Calculations Part 1

Scenario	Type	Calculation	Hand Calc	Spreadsheet
1	A - 1	S1:=Aggregate_DRH·Annual_Scalar	S1=217.7466 *cfs*	217.7466
2	A - 2	S2:=Aggregate_DRH·Monthly_Scalar	S2=5.1918 *cfs*	5.191794
3	B - 1	S3:=N1_DRH·Annual_Scalar	S3=168.6471 *cfs*	168.6472
4	B - 2	S4:=N1_DRH·Monthly_Scalar	S4=4.0211 *cfs*	4.021102
5	C - 1	S5:=NAverage_DRH·Annual_Scalar	S5=181.8169 *cfs*	185.131
6	C - 2	S6:=NAverage_DRH·Monthly_Scalar	S6=4.3351 *cfs*	4.41411
7	D - 3	S7:=N1_DailyFlow·DAR	S7=7.2922 *cfs*	7.29007
8	D - 4	S8:=N1_DailyFlow·AnnualAvgRatio	S8=7.1828 *cfs*	7.18279
9	D - 5	S9:=N1_DailyFlow·MonthlyAvgRatio	S9=7.0882 *cfs*	7.08822
10	D - 6	$S10 := \left(\left(N1\_DailyFlow - \mu_1 \right) \cdot \left(\frac{\sigma_2}{\sigma_1} \right) + \mu_2 \right)$	S10=11.8526 *cfs*	11.85257
Actual	Actual Daily Flow for Prediction Site:		PSite_DailyFlow=2.6 *cfs*	

Note: Standard Devation Equation (Traditional Method 3)

$$Q_2 = \left(Q_1 - \mu_1 \right) \cdot \left(\frac{\sigma_2}{\sigma_1} \right) + \mu_2$$

Q_1 Actual Daily Flow Values

σ_1 & σ_2 Monthly Standard Deviation

μ_1 & μ_2 Monthly Mean Flow Ratio

Figure 6 - Hand Calculations Part 2

Results

Results from each scenario were compared to the actual daily flows of the prediction site. Traditional scaling methods worked very well when reference gauges within the same class were chosen. Figure 7 on the following page shows all 10 scenario results for the 1969 water-year at USGS gauge 11264500. Note how the runoff peaks estimated by the DRH values were of a similar duration to the actual runoff, but shifted to earlier in the year (Scenarios 1 through 6). This shift is likely due to seasonal shifts made over the 20-year period used to define the DRH values.

Traditional methods were found to be very accurate at predicting the daily flows for the entire year. Such a high level of accuracy was assumed to correlate directly with the fact that all reference gauges were both near the prediction site, and of a similar class.

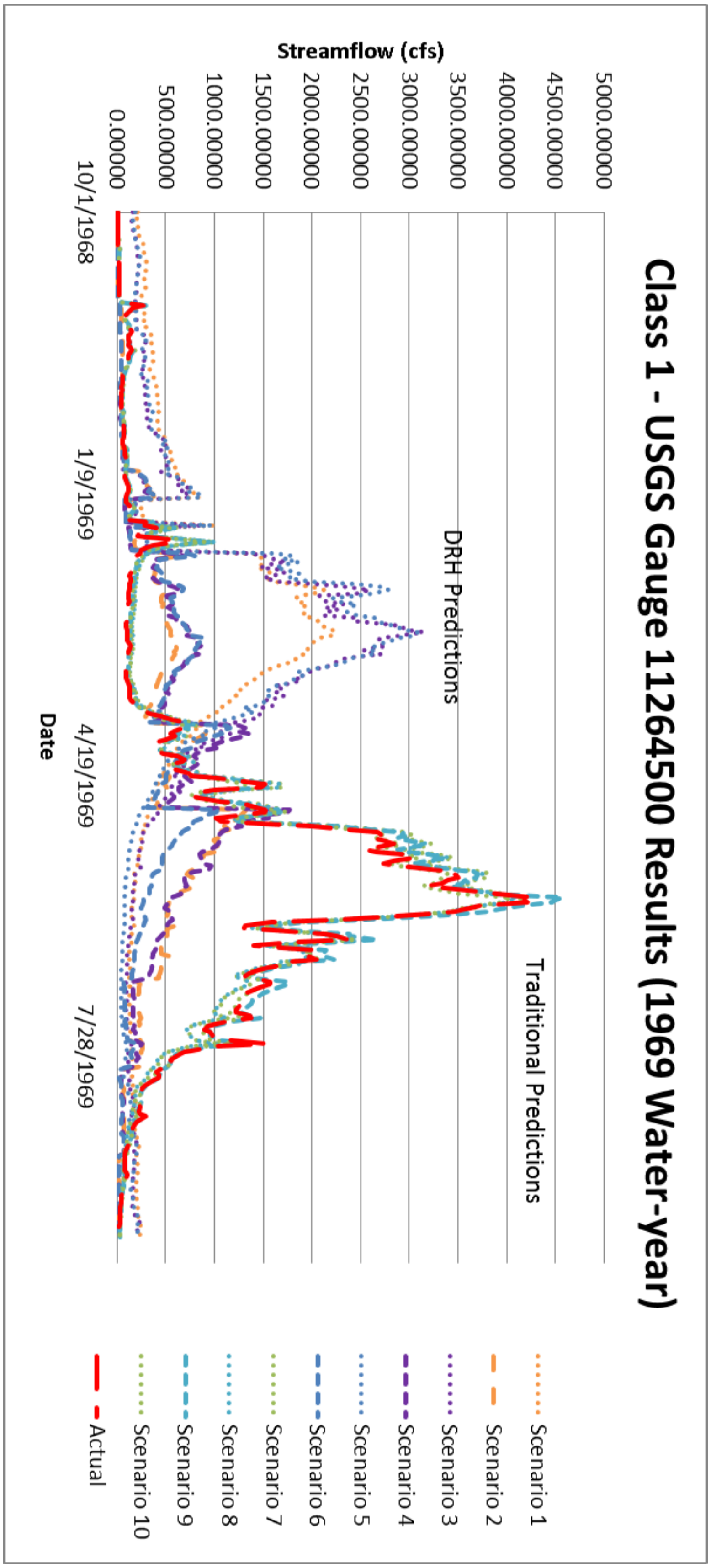

Figure 7 - USGS Gauge 11264500 Results

Limitations for the traditional scaling methods were found when predicting streamflow between gauges from different classes. For example, a stream that is primarily fed by snowmelt is shown by USGS Gauge 11264500. It has a hydrograph with one large peak from snowmelt runoff in the spring and relatively low flow in the late summer and fall (see Figure 8).

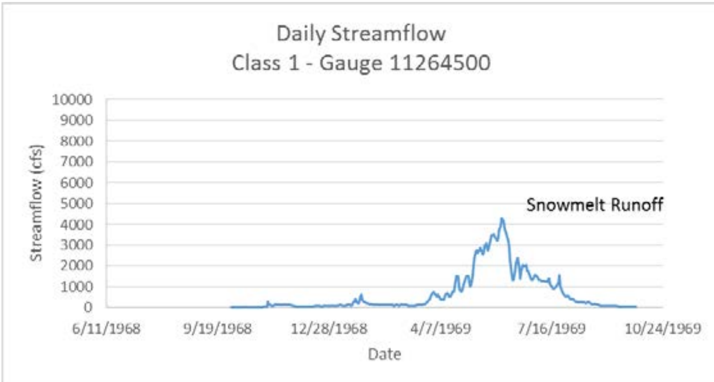

Figure 8 - Daily Streamflow Gauge 11264500

Gauge 11268000 shows a Class 3 stream fed by snowmelt and seasonal rainfall that has a runoff peak along with variable seasonal flows (see Figure 9).

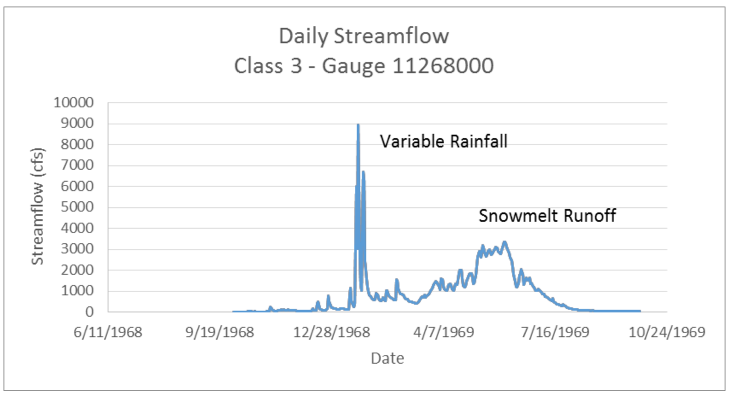

Figure 9 - Daily Streamflow Gauge 11268000

When the DAR method is used to estimate one year of daily flows for Gauge 11264500 by using Gauge 11268000 the shape of the hydrograph is incorrect (see Figure 10).

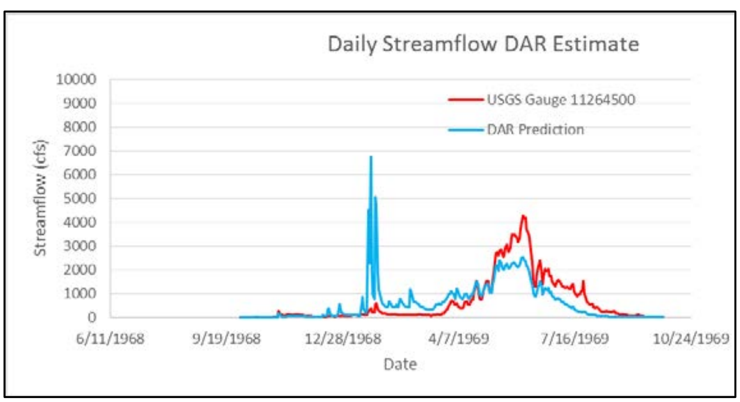

Figure 10 - Daily and Predicted Streamflow Gauge 11264500

Conclusion

The results discussed thus far show an indeterminate conclusion without further data. The DRH values did not produce more accurate results than the traditional methods but the traditional methods were only supremely accurate when used within a class. The classification metrics are more consistent at producing correctly shaped hydrographs over a series of years but lack accuracy for specific days. ArcGIS proved to be an invaluable resource in organizing the gauge data points and selecting the reference and prediction sites. While the original objective to create flow metrics for every stream in California was not met, the results provided will provide a strong basis and the tools necessary to create those metrics in the future.

Direction for Future Work

The ultimate goal of this project is to create an ArcGIS tool in python to calculate daily flows of ungauged sites. While this goal will be attainable in future efforts, the initial analysis of scaling methods were too time intensive to be completed within a single semester. Further work on this project will involve comparing additional results and choosing an optimal method or combination of methods to create satisfactory predictions. A fully debugged and user-friendly tool will likely not be completed until spring of next year. If successful, this tool could be used to predict flows across the state of California and serve as an example for other project areas.

REFERENCES

Chinnayakanahalli, K. J. (2010). "Characterizing Ecologically Relevant Variations in Streamflow Regimes." All Graduate Theses and Dissertations, Paper 561

Farmer, W. H., and Vogel, R. M. (2012). "Performance-Weighted Methods for Estimating Monthly Streamflow at Ungauged Sites ." Department of Civil and Environmental Engineering, 200 College Avenue, Medford, MA 02155, United States

Hersch, E. S., and Maidment, D. R. (2017). "An Integrated Stream Classification System for Texas." CENTER FOR RESEARCH IN WATER RESOURCES Bureau of Engineering Research • The University of Texas at Austin J.J. Pickle Research Campus • Austin, TX 78712-4497, <http://www.crwr.utexas.edu/online.shtml> (Oct. 1, 2017)

Lane, B. A., Dahlke, H. E., Pasternack, G. B., and Sandoval-Solis, S. (n.d.). "Revealing the diversity of Natural Hydrologic Regimes in California with Relevance for Environmental Flows Applications." Journal of the American Water Resources, 1–20

Poff, N.L., J.D. Allan, M.B. Bain, J.R. Karr, K.L. Prestegaard, B.D. Richter, R.E. Sparks, and J.C. Stromberg, 1997. The Natural Flow Regime. BioScience 47(11):769-784, DOI: 10.2307/1313099.

APPENDIX A

Visual Basic Code used to run each scenario in Excel

```
'       ################################                    MAIN SCENARIO               ################################

Sub Scenario(UI)
Dim StreamClass, ClassCount As Integer
Dim S, AM, RefType, N1, N2, N3, StartTime, EndTime, BeginWY As Double
Dim Psite, ClassNum As String
Dim Pws As Worksheet
Dim DRH, GageArea, SI, Class, AllFiles, AllFilesMatch, TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch As Variant

'Dim UI As Double        'Test
'UI = 9

StartTime = Round(Timer, 2)

' Retrieve Scenario Information
SI = Worksheets("Scenarios").Range("A3:I68")          'Array of Scenario Information
S = SI(UI, 1)                                          'Scenario Number
AM = SI(S, 2)                                          'Annual or Monthly
RefType = SI(S, 3)                                     'Type A, B, C, D, or Actual
StreamClass = SI(S, 4)                                 'Stream Class
ClassNum = "Class " & CStr(StreamClass)                'Create Class Number variable to call worksheet (Worksheet for each class must be prepared)
N1 = SI(S, 5)                                          'First Nearest Site
N2 = SI(S, 6)                                          'Second Nearest Site
N3 = SI(S, 7)                                          'Third Nearest Site
Psite = CStr(SI(S, 8))                                 'Prediction Site    (Worksheet for each prediction site must be prepared)
BeginWY = SI(S, 9)                                    'Beginning Water Year

' Gather Reference Data


Set Pws = ActiveWorkbook.Worksheets(Psite)                                          'Assign the worksheet for the prediction site
DRH = Worksheets("DRH").Range("A2:J366")                                            'DRH - Hydrograph for all Sites
GageArea = Worksheets("Gage_Areas").Range("A2:B366")                                'Gage Areas
Class = Worksheets(ClassNum).Range("A1:ZZ367")                                      'Unitless Hydrograph for Class Sites
ClassCount = Worksheets(ClassNum).Range("A1").Cells(1, Columns.Count).End(xlToLeft).Column          'Unitless Hydrograph Column
AllFiles = Worksheets("AllFiles").Range("A2:FY16074")                               'Daily Flows for all gages
AllFilesMatch = Worksheets("AllFiles").Range("A2:FY2").Value                        'Match Column for Daily Flows
AllFilesMatch = Application.Transpose(Application.Transpose(Worksheets("AllFiles").Range("A2:FY2").Value))

'  ################################              Prepare the Time Series Array

If RefType = "A" Then
    Call TimeSeries_A(DRH, StreamClass, TimeSeriesValue, TimeSeriesMatch)                        'Aggregate DRH Values

ElseIf RefType = "B" Then
    Call TimeSeries_B(DRH, N1, Class, ClassCount, TimeSeriesValue, TimeSeriesMatch)             'Nearest 1 DRH Values

ElseIf RefType = "C" Then
    Call TimeSeries_C(DRH, N1, N2, N3, Class, ClassCount, TimeSeriesValue, TimeSeriesMatch)     'Nearest 3 DRH Values

ElseIf RefType = "D" Then
    Call TimeSeries_D(AllFiles, AllFilesMatch, N1, TimeSeriesValue, TimeSeriesMatch, BeginWY)          'Daily flow values for Reference Gauge Site (N1)

ElseIf RefType = "Actual" Then
    Call TimeSeries_Actual(AllFiles, AllFilesMatch, Psite, TimeSeriesValue, TimeSeriesMatch, BeginWY)    'Daily flow values for Prediction Site

End If
  '  ################################              Prepare the Scalar
If AM = "Annual" Then
    Call Scalar_1(Pws, ScalarValue, ScalarMatch)                                  'Annual Scalar

ElseIf AM = "Monthly" Then
    Call Scalar_2(Pws, ScalarValue, ScalarMatch)                                 'Monthly Scalar

ElseIf AM = "DAR" Then
    Call Scalar_3(GageArea, N1, Psite, ScalarValue, ScalarMatch)                 'Direct Area Ratio Scalar

ElseIf AM = "AnnualAvg" Then
    Call Scalar_4(Pws, N1, AllFiles, AllFilesMatch, ScalarValue, ScalarMatch, BeginWY)

ElseIf AM = "MonthlyAvg" Then                                                    'Monthly Average Scalar
    Call Scalar_5(Pws, N1, AllFiles, AllFilesMatch, ScalarValue, ScalarMatch, BeginWY)

ElseIf AM = "StandDev" Then
    Call Scalar_6(Pws, N1, AllFiles, AllFilesMatch, ScalarValue, ScalarMatch, BeginWY)
End If
```

```vba
'       ##################################        Call Subroutines to Preform Calculations and Fill Results Table

If AM = "Annual" Then
    Call Annual(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
ElseIf AM = "Monthly" Then
    Call Monthly(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
ElseIf AM = "DAR" Then
    Call DARCalc(TimeSeriesValue, TimeSeriesMatch, ScalarValue, BeginWY, S)
ElseIf AM = "MonthlyAvg" Then
    Call MonthlyAverage(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
ElseIf AM = "AnnualAvg" Then
    Call AnnualAverage(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
ElseIf AM = "Actual" Then
    Call PrintActual(TimeSeriesValue, TimeSeriesMatch, BeginWY, S)
ElseIf AM = "StandDev" Then
    Call StandDev(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
End If


EndTime = Round(Timer, 2)


Worksheets("Scenarios").Range("J3").Cells(S, 1) = Time & "  " & Date
Worksheets("Scenarios").Range("K3").Cells(S, 1) = EndTime - StartTime


End Sub

' #################################              TIME SERIES           #################################


Sub TimeSeries_A(DRH, StreamClass, TimeSeriesValue, TimeSeriesMatch)
Dim i As Integer
ReDim TimeSeriesValue(365), TimeSeriesMatch(365)
For i = 1 To 365
    TimeSeriesValue(i) = DRH(i, StreamClass + 1)                        'Assign Aggregate DRH Values
    TimeSeriesMatch(i) = Month(DRH(i, 1)) & Day(DRH(i, 1))              'Assign Month and Day for each DRH Value
Next i

End Sub

Sub TimeSeries_B(DRH, N1, Class, ClassCount, TimeSeriesValue, TimeSeriesMatch)
Dim N1Column, i As Integer
ReDim TimeSeriesValue(365), TimeSeriesMatch(365)

N1Column = FindColumn(N1, Class, ClassCount)

For i = 1 To 365
    TimeSeriesValue(i) = Class(i + 2, N1Column)                       'Assign Nearest 1 DRH Values
    TimeSeriesMatch(i) = Month(DRH(i, 1)) & Day(DRH(i, 1))            'Assign Month and Day for each DRH Value
Next i

End Sub

Sub TimeSeries_C(DRH, N1, N2, N3, Class, ClassCount, TimeSeriesValue, TimeSeriesMatch)
Dim N1Column, N2Column, N3Column, i As Integer
ReDim TimeSeriesValue(365), TimeSeriesMatch(365)

N1Column = FindColumn(N1, Class, ClassCount)
N2Column = FindColumn(N2, Class, ClassCount)
N3Column = FindColumn(N3, Class, ClassCount)


For i = 1 To 365
    TimeSeriesValue(i) = (Class(i + 2, N1Column) + Class(i + 2, N2Column) + Class(i + 2, N3Column)) / 3
    TimeSeriesMatch(i) = Month(DRH(i, 1)) & Day(DRH(i, 1))
Next i

End Sub
```

```
Sub TimeSeries_D(AllFiles, AllFilesMatch, N1, TimeSeriesValue, TimeSeriesMatch, BeginWY)
Dim i, j, N1Column, N1RowBegin, N1RowEnd As Integer
Dim AllFilesCount As Double
Dim BeginDate, EndDate As String
ReDim TimeSeriesValue(7310), TimeSeriesMatch(7310)


AllFilesCount = Worksheets("AllFiles").Range("A2").Cells(1, Columns.Count).End(xlToLeft).Column
N1Column = FindColumn(N1, AllFiles, AllFilesCount)

If BeginWY = 1969 Then
    BeginDate = "10/1/1968"
    EndDate = "9/30/1988"
    N1RowBegin = 2
    N1RowEnd = 7306
ElseIf BeginWY = 1990 Then
    BeginDate = "10/1/1989"
    EndDate = "9/30/2009"
    N1RowBegin = 7307
    N1RowEnd = 14976
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If

j = 0
For i = N1RowBegin To N1RowEnd
    j = j + 1
    TimeSeriesValue(j) = AllFiles(i, N1Column)
    TimeSeriesMatch(j) = AllFiles(i, 1)
Next i


End Sub

Sub TimeSeries_Actual(AllFiles, AllFilesMatch, Psite, TimeSeriesValue, TimeSeriesMatch, BeginWY)
Dim PColumn, i, j As Integer
Dim PSiteInt, AllFilesCount, N1RowBegin, N1RowEnd As Double
ReDim TimeSeriesValue(7306), TimeSeriesMatch(7306)

If BeginWY = 1969 Then
    N1RowBegin = 2
    N1RowEnd = 7306
ElseIf BeginWY = 1990 Then
    N1RowBegin = 7307
    N1RowEnd = 14976
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If

PSiteInt = CDbl(Psite)
AllFilesCount = Worksheets("AllFiles").Range("A2").Cells(1, Columns.Count).End(xlToLeft).Column
PColumn = FindColumn(PSiteInt, AllFiles, AllFilesCount)

j = 0
For i = N1RowBegin To N1RowEnd
    j = j + 1
        TimeSeriesValue(j) = AllFiles(i, PColumn)
        TimeSeriesMatch(j) = AllFiles(i, 1)
Next i

End Sub
```

```
' ################################                SCALARS           ################################

Sub Scalar_1(Pws, ScalarValue, ScalarMatch) 'Annual Average
Dim i As Integer
ReDim ScalarMatch(65)
ReDim ScalarValue(65)


For i = 1 To 65
    ScalarMatch(i) = Pws.Range("L2").Cells(i, 1)
    ScalarValue(i) = Pws.Range("L2").Cells(i, 2)
Next i

End Sub

Sub Scalar_2(Pws, ScalarValue, ScalarMatch) 'Monthly Average
Dim i As Integer
ReDim ScalarMatch(792), ScalarValue(792)



For i = 1 To 792
    ScalarMatch(i) = Pws.Range("N2").Cells(i, 1)
    ScalarValue(i) = Pws.Range("N2").Cells(i, 2)
Next i

End Sub

Sub Scalar_3(GageArea, N1, Psite, ScalarValue, ScalarMatch) 'DAR
Dim PSiteInt, i As Integer
Dim N1_Area, PSite_Area As Double

    PSiteInt = CDbl(Psite)
    For i = 1 To 207
        If GageArea(i, 1) = N1 Then
            N1_Area = GageArea(i, 2)
        End If
        If GageArea(i, 1) = PSiteInt Then
            PSite_Area = GageArea(i, 2)
        End If
    Next i
    If N1_Area = 0 Then
        MsgBox ("Nearest 1 Area: " & N1 & " Not Found")                 'Warn User if Nearest 1 Area  Not Found
        Exit Sub                                                         'Exit Sub if Nearest 1 Area  Not Found
    ElseIf PSite_Area = 0 Then
        MsgBox ("Prediction Site Area: " & Psite & " Not Found")         'Warn User if Prediction Site Area Not Found
        Exit Sub                                                         'Exit Sub if Prediction Site Area Not Found
    End If

    ScalarValue = PSite_Area / N1_Area                                   'Calculate Drainage Area Ratio (DAR) Value


End Sub
```

```
Sub Scalar_4(Pws, N1, AllFiles, AllFilesMatch, ScalarValue, ScalarMatch, BeginWY) 'Annual Average Ratio
Dim i, j, k, yr, mnth, yr_old, N1Column As Integer
Dim PSite_AnnualAvg, N1_AnnualAvg(20), N1_Sum, AnnualAvgRatio(20), N1RowBegin, N1RowEnd, AllFilesCount As Double
ReDim ScalarValue(20), ScalarMatch(20)

If BeginWY = 1969 Then
    N1RowBegin = 2
    N1RowEnd = 7306
ElseIf BeginWY = 1990 Then
    N1RowBegin = 7307
    N1RowEnd = 14976
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


AllFilesCount = Worksheets("AllFiles").Range("A2").Cells(1, Columns.Count).End(xlToLeft).Column
N1Column = FindColumn(N1, AllFiles, AllFilesCount)
j = 0
yr_old = 0

For i = 2 To 7306                                                       'Loop through N1 Column of All Files
    If AllFiles(i, N1Column) = "NA" Then
        GoTo Scalar_4SkipIteration
    Else
        yr = Year(AllFiles(i, 1))
        mnth = Month(AllFiles(i, 1))

        If mnth = 10 Or mnth = 11 Or mnth = 12 Then
            yr = yr + 1                                                     'Use Water Year
        End If

        If yr > yr_old Then
            j = j + 1
            k = 0
            N1_Sum = 0
        End If
        k = k + 1                                                           'Days in Year
        N1_Sum = (N1_Sum + AllFiles(i, N1Column))                          'Annual Average
        N1_AnnualAvg(j) = N1_Sum / k                                        'Assign Annual Average for Year
        yr_old = yr
    End If
Scalar_4SkipIteration:
Next i

For i = 1 To 20                                                         'Only use 20 years because the daily data only goes to 20 years
    If N1_AnnualAvg(i) > 0 Then
        PSite_AnnualAvg = Pws.Range("L2").Cells(i, 2)
        ScalarValue(i) = PSite_AnnualAvg / N1_AnnualAvg(i)                  'Annual Ratio for given Water Year
        ScalarMatch(i) = 1968 + i                                           'Water Year Match Value
    End If
Next i

End Sub
```

```vba
Sub Scalar_5(Pws, N1, AllFiles, AllFilesMatch, ScalarValue, ScalarMatch, BeginWY)
Dim i, mnth, yr, N1Column As Integer
Dim N1RowBegin, N1RowEnd, AllFilesCount As Double
Dim PredictedMonthlyValue, PredictedMatch, PSiteMonthlyAvg, N1MonthlyAvg  As Variant

If BeginWY = 1969 Then
    N1RowBegin = 2
    N1RowEnd = 7306
ElseIf BeginWY = 1990 Then
    N1RowBegin = 7307
    N1RowEnd = 14976
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


PredictedMonthlyValue = Pws.Range("E227:E466")                          'PSite Values
PredictedMatch = Pws.Range("C227:D466")                                 'PSite Months

ReDim PSiteMonthlyAvg(3, 12)                                       '
ReDim N1MonthlyAvg(3, 12)


For i = 1 To 240                                                                  'Sum Prediction Site Monthly Flows, and correlating number of months
    mnth = PredictedMatch(i, 2)                                                   'Month
    yr = PredictedMatch(i, 1)                                                     'Year
    If mnth = 10 Or mnth = 11 Or mnth = 12 Then
        yr = yr + 1
    End If
    PSiteMonthlyAvg(1, mnth) = PSiteMonthlyAvg(1, mnth) + PredictedMonthlyValue(i, 1)    'Sum each monthly value
    PSiteMonthlyAvg(2, mnth) = PSiteMonthlyAvg(2, mnth) + 1                       'Number of months
    PSiteMonthlyAvg(3, mnth) = PSiteMonthlyAvg(1, mnth) / PSiteMonthlyAvg(2, mnth) 'Average
Next i

AllFilesCount = Worksheets("AllFiles").Range("A2").Cells(1, Columns.Count).End(xlToLeft).Column
N1Column = FindColumn(N1, AllFiles, AllFilesCount)                                        'Find column for nearest gage

For i = N1RowBegin To N1RowEnd
    mnth = Month(AllFiles(i, 1))
    If AllFiles(i, N1Column) = "NA" Then
        GoTo Scalar_5_nexti
    End If
    N1MonthlyAvg(1, mnth) = N1MonthlyAvg(1, mnth) + AllFiles(i, N1Column)         'Sum each daily value
    N1MonthlyAvg(2, mnth) = N1MonthlyAvg(2, mnth) + 1                             'Number of days
    N1MonthlyAvg(3, mnth) = N1MonthlyAvg(1, mnth) / N1MonthlyAvg(2, mnth)         'Average
Scalar_5_nexti:
Next i

ReDim ScalarValue(12)
ReDim ScalarMatch(12)

For i = 1 To 12
    ScalarValue(i) = PSiteMonthlyAvg(3, i) / N1MonthlyAvg(3, i)
    ScalarMatch(i) = i
Next i

End Sub
```

```
Sub Scalar_6(Pws, N1, AllFiles, AllFilesMatch, ScalarValue, ScalarMatch, BeginWY)
Dim i, j, yr, yr_old, mnth, N1Column As Integer
Dim PredictedMonthlyValue, PredictedMatch, PSiteMonthlyValue, N1MonthlyValue, PSiteMonthlyAvg, N1MonthlyAvg, N1Monthly, PSiteMonthly As Variant
Dim PSiteStandDev, N1StandDev, N1_Avg, PSite_Avg, SD, ED, N1RowBegin, N1RowEnd, AllFilesCount As Double
Dim StartDate, EndDate As Date

If BeginWY = 1969 Then
    N1RowBegin = 2
    N1RowEnd = 7306
ElseIf BeginWY = 1990 Then
    N1RowBegin = 7307
    N1RowEnd = 14976
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


'Create array of monthly averages
PredictedMonthlyValue = Pws.Range("E227:E466")                          'PSite Values
PredictedMatch = Pws.Range("C227:D466")                                 'PSite Months

ReDim PSiteMonthlyAvg(1, 12)
ReDim N1MonthlyAvg(1, 12)

ReDim PSiteMonthly(12, 20)


For i = 1 To 240                                                       'Sum Prediction Site Monthly Flows, and correlating number of months
    mnth = PredictedMatch(i, 2)                                        'Month
    yr = PredictedMatch(i, 1)                                          'Year
    If mnth = 10 Or mnth = 11 Or mnth = 12 Then                        'Convert to Water Year
        yr = yr + 1
    End If
    j = yr - (BeginWY - 1)                                                 'Row based on year for the array
    PSiteMonthly(mnth, j) = PredictedMonthlyValue(i, 1)              'Assign each monthly value
Next i

AllFilesCount = Worksheets("AllFiles").Range("A2").Cells(1, Columns.Count).End(xlToLeft).Column
N1Column = FindColumn(N1, AllFiles, AllFilesCount)                              'Find column for nearest gage

yr_old = 0
ReDim N1Monthly(12, 20)
For i = N1RowBegin To N1RowEnd
    mnth = Month(AllFiles(i, 1))
    yr = Year(AllFiles(i, 1))
    If mnth = 10 Or mnth = 11 Or mnth = 12 Then                        'Convert to Water Year
        yr = yr + 1
    End If

    If yr > yr_old Then
        Erase N1MonthlyAvg                                                         'Clear Months each new water year
        ReDim N1MonthlyAvg(3, 12)
    End If

    N1MonthlyAvg(1, mnth) = N1MonthlyAvg(1, mnth) + AllFiles(i, N1Column)                'Sum each daily value
    N1MonthlyAvg(2, mnth) = N1MonthlyAvg(2, mnth) + 1                                    'Number of days
    N1MonthlyAvg(3, mnth) = N1MonthlyAvg(1, mnth) / N1MonthlyAvg(2, mnth)                'Average

    j = yr - 1968
    N1Monthly(mnth, j) = N1MonthlyAvg(3, mnth)

    yr_old = yr

Next i
'Find Standard Deviation for each month
ReDim ScalarMatch(12)
ReDim ScalarValue(12, 4)

For mnth = 1 To 12

    ScalarMatch(mnth) = mnth                                 'Match Value is the month

    N1_Avg = Application.Average(Application.Index(N1Monthly, mnth, 0))
    ScalarValue(mnth, 1) = StdDev(N1Monthly, mnth, N1_Avg)              'Standard Deviation for N1

    PSite_Avg = Application.Average(Application.Index(PSiteMonthly, mnth, 0))
    ScalarValue(mnth, 2) = StdDev(PSiteMonthly, mnth, PSite_Avg)         'Standard Deviation for PSite

    ScalarValue(mnth, 3) = N1_Avg               'Monthly Average for N1
    ScalarValue(mnth, 4) = PSite_Avg            'Monthly Average for PSite

Next mnth


End Sub
```

```
'   ################################           OUTPUT           ################################

Sub Annual(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
Dim StartDate, EndDate As Date
Dim j, i, yr, mnth, dy As Integer
Dim SearchDay As String
Dim DDRH, DScalar As Variant

If BeginWY = 1969 Then
    StartDate = "10/1/1968"
    EndDate = "9/30/1988"
    j = 0
ElseIf BeginWY = 1990 Then
    StartDate = "10/1/1989"
    EndDate = "9/30/2009"
    j = 7306
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


For i = StartDate To EndDate                                        'Loop Through each day
    j = j + 1                                                       'For Cell Count in output

    With Application
        yr = Year(i)
        mnth = Month(i)
        dy = Day(i)
        If mnth = 10 Or mnth = 11 Or mnth = 12 Then                          'Convert year to water year
            yr = yr + 1
        End If
        If mnth = 2 And dy = 29 Then
            GoTo NextIteration                                                 'Skip Leap Days
        End If
        SearchDay = mnth & dy                                                  'Value used to find the month & day of interest in the DRH (independent of year)

        DDRH = .Index(TimeSeriesValue, .Match(SearchDay, TimeSeriesMatch, 0))    'Daily DRH Value
        DScalar = .Index(ScalarValue, .Match(yr, ScalarMatch, 0))              'Yearly Scalar Value
    End With

    Worksheets("Results").Range("B2").Cells(j, S) = DDRH * DScalar

NextIteration:
Next i

End Sub

Sub Monthly(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
Dim StartDate, EndDate As Date
Dim j, i, yr, mnth, dy As Integer
Dim SearchDay As String
Dim DDRH, DScalar As Variant

If BeginWY = 1969 Then
    StartDate = "10/1/1968"
    EndDate = "9/30/1988"
    j = 0
ElseIf BeginWY = 1990 Then
    StartDate = "10/1/1989"
    EndDate = "9/30/2009"
    j = 7306
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


For i = StartDate To EndDate                                        'Loop Through each day
    j = j + 1                                                       'For Cell Count in output

    With Application
        yr = Year(i)
        mnth = Month(i)
        dy = Day(i)
        If mnth = 2 And dy = 29 Then
            GoTo NextIteration                                                 'Skip Leap Days
        End If
        SearchDay = mnth & dy                                                  'Value used to find the month & day of interest in the DRH (independent of year)

        DDRH = .Index(TimeSeriesValue, .Match(SearchDay, TimeSeriesMatch, 0))        'Daily DRH Value
        DScalar = .Index(ScalarValue, .Match(yr & mnth, ScalarMatch, 0))       'Monthly Scalar Value
    End With

    Worksheets("Results").Range("B2").Cells(j, S) = DDRH * DScalar

NextIteration:
Next i

End Sub
```

```
Sub DARCalc(TimeSeriesValue, TimeSeriesMatch, DAR, BeginWY, S)
Dim StartDate, EndDate As Date
Dim i, j As Integer
Dim SearchDay As String
Dim DailyValue As Variant

If BeginWY = 1969 Then
    StartDate = "10/1/1968"
    EndDate = "9/30/1988"
    j = 0
ElseIf BeginWY = 1990 Then
    StartDate = "10/1/1989"
    EndDate = "9/30/2009"
    j = 7306
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


For i = StartDate To EndDate                                    'Loop Through each day
    j = j + 1                                                   'For Cell Count in output
    With Application
    DailyValue = .Index(TimeSeriesValue, .Match(i, TimeSeriesMatch, 0))     'Daily Value
    End With
    If DailyValue = "NA" Then
    GoTo GoNext
    End If
    Worksheets("Results").Range("B2").Cells(j, S) = DailyValue * DAR
GoNext:
Next i

End Sub

Sub PrintActual(TimeSeriesValue, TimeSeriesMatch, BeginWY, S)
Dim StartDate, EndDate As Date
Dim i, j As Integer
Dim DailyValue As Variant

If BeginWY = 1969 Then
    StartDate = "10/1/1968"
    EndDate = "9/30/1988"
    j = 0
ElseIf BeginWY = 1990 Then
    StartDate = "10/1/1989"
    EndDate = "9/30/2009"
    j = 7306
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


For i = StartDate To EndDate                                    'Loop Through each day
    j = j + 1                                                   'For Cell Count in output
    With Application
    Worksheets("Results").Range("B2").Cells(j, S) = .Index(TimeSeriesValue, .Match(i, TimeSeriesMatch, 0))
    End With
Next i

End Sub
```

```
Sub MonthlyAverage(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
Dim StartDate, EndDate As Date
Dim MScalar, DailyValue As Double
Dim i, j, mnth As Integer

If BeginWY = 1969 Then
    StartDate = "10/1/1968"
    EndDate = "9/30/1988"
    j = 0
ElseIf BeginWY = 1990 Then
    StartDate = "10/1/1989"
    EndDate = "9/30/2009"
    j = 7306
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If

For i = StartDate To EndDate                                'Loop Through each day
    j = j + 1                                               'For Cell Count in output

    With Application
        mnth = Month(i)                                                 'Month
        DailyValue = .Index(TimeSeriesValue, .Match(i, TimeSeriesMatch, 0))   'Daily Value from Nearest
        MScalar = .Index(ScalarValue, .Match(mnth, ScalarMatch, 0))     'Monthly Scalar
    End With

    Worksheets("Results").Range("B2").Cells(j, S) = DailyValue * MScalar

Next i

End Sub

Sub AnnualAverage(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
Dim StartDate, EndDate As Date
Dim j, i, yr, mnth As Integer
Dim SearchDay, AScalar, DailyValue, MonthlyValue As Double

If BeginWY = 1969 Then
    StartDate = "10/1/1968"
    EndDate = "9/30/1988"
    j = 0
ElseIf BeginWY = 1990 Then
    StartDate = "10/1/1989"
    EndDate = "9/30/2009"
    j = 7306
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If
                                                       'Reference for output cell row
For i = StartDate To EndDate                                  'Loop Through each day
    j = j + 1                                                 'For Cell Count in output
    yr = Year(i)
    mnth = Month(i)
    If mnth = 10 Or mnth = 11 Or mnth = 12 Then                   'Convert year to water year
        yr = yr + 1
    End If
    With Application
        AScalar = .Index(ScalarValue, .Match(yr, ScalarMatch, 0))            'Annual Scalar
        DailyValue = .Index(TimeSeriesValue, .Match(i, TimeSeriesMatch, 0))  'Daily Value

    End With
    If DailyValue <> "NA" Then
        Worksheets("Results").Range("B2").Cells(j, S) = DailyValue * AScalar
    End If

Next i

End Sub
```

```
Sub StandDev(TimeSeriesValue, TimeSeriesMatch, ScalarValue, ScalarMatch, BeginWY, S)
Dim StartDate, EndDate As Date
Dim j, i, yr, mnth As Integer
Dim dev1, dev2, mean1, mean2, Q1 As Double

If BeginWY = 1969 Then
    StartDate = "10/1/1968"
    EndDate = "9/30/1988"
    j = 0
ElseIf BeginWY = 1990 Then
    StartDate = "10/1/1989"
    EndDate = "9/30/2009"
    j = 7306
Else
    MsgBox ("Beginning WY incorrect")
    Exit Sub
End If


For i = StartDate To EndDate                                    'Loop Through each day
    j = j + 1                                                   'For Cell Count in output
    yr = Year(i)
    mnth = Month(i)
    If mnth = 10 Or mnth = 11 Or mnth = 12 Then                       'Convert year to water year
            yr = yr + 1
    End If
    With Application
        Q1 = .Index(TimeSeriesValue, .Match(i, TimeSeriesMatch, 0))      'Daily Value
    End With

    dev1 = ScalarValue(mnth, 1)
    dev2 = ScalarValue(mnth, 2)
    mean1 = ScalarValue(mnth, 3)
    mean2 = ScalarValue(mnth, 4)


    Worksheets("Results").Range("B2").Cells(j, S) = (Q1 - mean1) * (dev2 / dev1) + mean2           'Q2 in the equation


Next i

End Sub
```

```vba
' ################################          Functions          ################################



Function FindColumn(NValue, Table, Columns) As Integer
Dim NColumn, i As Integer

NColumn = 0
For i = 1 To Columns
    If Table(1, i) = NValue Then
        FindColumn = i                                              'Find Column in Allfiles Worksheet
    End If
Next i
If FindColumn = 0 Then
    MsgBox ("Nearest Value: " & NValue & " Not Found")               'Warn User if Value Not Found
    Exit Function                                                  'Exit Sub if Value Not Found
End If

End Function


Function StdDev(Arr, mnth, Avg)
Dim i As Integer
Dim SumSq As Single
Dim n As Long

n = 0

For i = 1 To 20
   n = n + 1
    SumSq = SumSq + (Arr(mnth, i) - Avg) ^ 2
Next i

StdDev = Sqr(SumSq / (n - 1))

End Function

Function CountRows(WSNAME, StartRow)
Set WS = ActiveWorkbook.Worksheets(WSNAME)
CountRows = WS.Cells(Rows.Count, 1).End(xlUp).Row

End Function

Function CountColumns(WSNAME, StartColumn) As Variant
Dim WS As Worksheet
Set WS = ActiveWorkbook.Worksheets(WSNAME)
CountColumns = WS.Cells(1, Columns.Count).End(xlUp).Column

End Function
```