Appendix C

filter.ncl

```
;*************************************************************************************
*********
; This program uses a Lanczos filter to calculate 1-D filter weights for the raw
temperature data.
; A weighted running average is then calculated using wgt_runave.
;*************************************************************************************
*********

;----------------------------------------------------------------
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/shea_util.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
;----------------------------------------------------------------

begin

 files=30                      ; 30 files representing the 30 stations
 num_time=10135                 ; 10135 times
 num_col=3                      ; column 1 = time, 2 = maximum, 3 = minimum
 nwt = 51                       ; Total number of weights
 fca = 0.333                    ; Cut-off frequency of the ideal low-pass filter
 ihp = 0                        ; For Low pass filter, ihp = 0
 nsigma = 1.                    ; Power of the sigma factor (nsigma >= 0), nsigma=1
common
 opt = 0                        ; End-point option (opt=0 most common)

 wgt          = new((/files,nwt/),float)            ; filter weights array
 filtered_max = new((/files,num_time/),float)              ; filtered maximum
temperatures array
 filtered_min = new((/files,num_time/),float)              ; filtered minimum
temperatures array
 filtered     = new((/files,num_time,num_col/),float)   ; 3-D filtered array of time,
max, and min filtered temperatures

 datafiles = (/"Redding", "RedBluff", "Colusa", "Sac", "Stockton", "Modesto", "Merced",
"Fresno", "Visalia", "Bkrsfld" \
             , "CrescentCity", "Eureka", "Covelo", "Graton", "SanFrancisco", "Monterey",
"SLO", "SantaBarbara", "SantaAna" \
             , "Vista", "Seattle","Spokane", "Yakima", "Portland", "Eugene", "Medford",
"Pendleton", "BakerCity", "Reno"  \
             , "Tonopah"/)

 do df=0,files-1
  print(df)
  data_filter=asciiread("IPM_"+datafiles(df)+".csv",(/num_time,num_col/),"float")  ; Read
in raw temperature data

  time      = data_filter(:,0)                       ; column 0 = time(date)
  maximum   = data_filter(:,1)                       ; column 1 = maximum daily temp
  minimum   = data_filter(:,2)                       ; column 2 = minimum daily temp

  wgt(df,:) = filwgts_lancos (nwt, ihp, fca, -999., nsigma) ; Lanczos filter function,
Calculates 1-D filter weights.

  filtered_max(df,:) = wgt_runave(maximum, wgt(df,:), opt)  ; Wgt_runave function
  filtered_min(df,:) = wgt_runave(minimum, wgt(df,:), opt)  ; Wgt_runave function

                                          ; Loop through to fill in the nwt missing
values with unfiltered raw data
                                          ; These dates will include the first and the
last nwt-1/2 values,
                                          ; which would be January 1979 and September
2007 temperatures
  do i=0,num_time-1
```

```
   if (ismissing(filtered_max(df,i)) .eq. True) then
    filtered_max(df,i) = maximum(i)
   end if
   if (ismissing(filtered_min(df,i)) .eq. True) then
    filtered_min(df,i) = minimum(i)
   end if
  end do

  filtered(df,:,0) = time
  filtered(df,:,1) = filtered_max(df,:)
  filtered(df,:,2) = filtered_min(df,:)

  asciiwrite(datafiles(df)+"_filtered_max", filtered(df,:,1))              ; write out
the filtered max temps to a file
  asciiwrite(datafiles(df)+"_filtered_min", filtered(df,:,2))              ; write out
the filtered min temps to a file

  opt1       = True                              ; File outs arrays to separate files, to
be called in from other programs
  opt1@fout  = datafiles(df)+"_filtered"
  write_matrix(filtered(df,:,:), "f8.2", opt1)
  print("opt1 fout done")

 end do

 print(filtered)

 print(filtered_max)
 print(filtered_min)

end
```

IPM.ncl

```
;**********************************************************************************
; This program separates the 01-01-1979 to 09-30-2006 JJAS heat wave months from
; the rest of the data set, creates daily temperature anomalies, and plots various
; statistics concerning the dataset.
;**********************************************************************************

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/shea_util.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

;----------------------------------------------------------------
begin

 num_time=10135                                        ; number of dates from 01-01-1979 to
09-30-2006
 num_col=3                                      ; 3 columns: 1)date, 2)maximum daily temp,
3)minimum daily temp
 files=30                                       ; Number of city files, 1 file for each city
 JJAS_days=3416                                        ; Number of days in all JJAS, 1 June
1979 to 30 Sept 2006
 yrs=28                                         ; Number of years
 dys=122                                               ; Number of days in June, July,
August, Sept

 datafiles = (/"Redding", "RedBluff", "Colusa", "Sac", "Stockton", "Modesto", "Merced",
"Fresno", "Visalia", "Bkrsfld" \
             , "CrescentCity", "Eureka", "Covelo", "Graton", "SanFrancisco", "Monterey",
"SLO", "SantaBarbara", "SantaAna" \
             , "Vista", "Seattle", "Spokane", "Yakima", "Portland", "Eugene", "Medford",
"Pendleton", "BakerCity", "Reno" \
             , "Tonopah"/)

 JJAS_time               = new((/files,JJAS_days/),float)   ; 2-D array of all JJAS
times, 3416 days in all JJAS 1979-2006
 JJAS_date_array         = new((/files,JJAS_days/),integer) ; 2-D array of all JJAS dates
 JJAS_max                = new((/files,JJAS_days/),float)   ; 2-D array of all JJAS
maximum temps
 JJAS_min                = new((/files,JJAS_days/),float)   ; 2-D array of all JJAS
minimum temps
 tmin_avg               = new((/files/),float)             ; 1-D array of Minimum
Temperature averages, for ALL JJAS
 tmax_avg                = new((/files/),float)                  ; 1-D array of Maximum
Temperature averages, for ALL JJAS
 rc_tmin                 = new((/files/),float)                  ; 1-D array of Min Temp
regression coefficients
 tmin_reg               = new((/files,JJAS_days/),float)   ; 2-D array of Tmin regression
coefficients
 data_reg                = new((/files,2,num_time/),float)  ; 3-D array of regression
data values
 max_array               = new((/files,yrs,dys/),float)     ; 3-D array of all JJAS
maximum temperatures
 min_array              = new((/files,yrs,dys/),float)     ; 3-D array of all JJAS minimum
temperatures
 max_array_time          = new((/files,yrs,dys/),float)     ; 3-D array of all JJAS times
 max_array_date          = new((/files,yrs,dys/),integer)   ; 3-D array of all JJAS dates
= 28 years,122 days in each yr
 over_max_array          = new((/files,yrs,dys/),float)     ; 3-D array of JJAS max temps
over 100
 LTDM                    = new((/files,dys/),float)         ; The Long Term Daily Mean
for each JJAS day, 1 June-30 Sept
 LTDM_extended           = new((/files,JJAS_days/),integer)
 anomaly                 = new((/files,yrs,dys/),float)     ; 3-D array of all JJAS
Anomalies
 over_anom               = new((/files,yrs,dys/),float)     ; 3-D array of anomalies that
exceed the Max anomaly threshold
 over_anom_date          = new((/files,yrs,dys/),integer)   ; 3-D array of dates of anoms
exceeding the Max anom threshold
```

```
 start_date              = new((/files,yrs,dys/),integer)   ; 3-D array of event starting
dates
 end_date                = new((/files,yrs,dys/),integer)   ; 3-D array of event ending
dates
 sum_max                 = new((/files,yrs,dys/),float)     ; 3-D array of the sum of the
event temps
 ave_max                 = new((/files,yrs,dys/),float)     ; 3-D array of the event
average maximum temps
 num_days                = new((/files,yrs,dys/),integer)   ; 3-D array of the # of days
in event with that start date
 indicator               = new((/files,yrs,dys/),float)     ; 3-D array indicates(0/1)
max temp exceeding max anom thrshld
 sum_test                = new((/files,yrs,dys/),float)     ; 3-D array of sums of # days
in event exceeding max anom thrshld
 event_start             = new((/files,yrs,dys/),integer)   ; 3-D array of event starting
dates
 num_days_events         = new((/files,yrs,dys/),integer)   ; 3-D array of the number of
days in each event
 event_max               = new((/files,yrs,dys/),float)     ; 3-D array of event max
Tmaxs
 event_min               = new((/files,yrs,dys/),float)     ; 3-D array of event min
Tmins
 b_max                   = new((/files/),integer)       ; 1-D array of number of events
for each city
 j_max                   = new((/files/),integer)       ; 1-D array of the max number of
event days
 event_num               = new((/files/),integer)       ; 1-D array of number of events
for each city
 days                    = new((/files/),integer)       ; 1-D array of the number of
event days for each city
 g_total                 = new((/files,yrs/),integer)           ; 2-D array of the number
of events each year for each city
 max_anom_ave            = new((/files,yrs,dys/),float)         ; 3-D array of the
maximum N-day anomaly average of each event
 max_anom_ave_date     = new((/files,yrs,dys/),integer)   ; 3-D array of the start date
of the max N-day anom avg
 max_temp              = new((/files,yrs,dys/),float)
 JJAS_SD                 = new((/files/),float)                 ; 1-D array of the
standard deviation of JJAS for each city
 norm_anom               = new((/files,yrs,dys/),float)         ; 3-D array of the daily
normalized anomalies
 norm_anom_events        = new((/files,yrs,dys/),float)         ; 3-D array of the
normalized anoms for event days
 norm_anom_avg           = new((/files/),float)                 ; 1-D array of the avg of
the norm event anoms for each city
 norm_anom_avg_all     = new((/files/),float)                 ; 1-D array of the avg of
all normalized anomalies

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

                                          ; THE Do Loop of all cities
; do df=0,files-1                         ; THE Do Loop of all cities, using
UNfiltered temperature data
;  print(df)
;  data=asciiread("IPM_"+datafiles(df)+".csv",(/num_time,num_col/),"float")       ; reads
in .csv file with temp data

 do df=0,files-1                          ; THE Do Loop of all cities, using FILTERED
temperature data
  print(df)
  data=asciiread(datafiles(df)+"_filtered",(/num_time,num_col/),"float")

  time      = data(:,0)                           ; column 0 = time(date)
  maximum   = data(:,1)                           ; column 1 = maximum daily temp
  minimum   = data(:,2)                           ; column 2 = minimum daily temp

printVarSummary(time)
  time!0         = "time"
```

```
  time&time      = time
  time@long_name = "date"
  time@units = "days after 1979-01-01 00:00:00"                    ; for example,
time(1979-01-01) = 0

  utc_date  = ut_calendar(time, 0)                            ; ut_calendar option 0 =
dimsizes(time) x 6:
  JJAS      = ut_calendar(time, 0)                            ; utc_date(:,0) --> years
  JJAS_date = ut_calendar(time, -2)                           ; utc_date(:,1) --> months
                                                              ; utc_date(:,2) --> days
                                                              ; utc_date(:,3) --> hours
                                                              ; utc_date(:,4) --> minutes
                                                              ; utc_date(:,5) --> seconds
                                                 ; ut_calendar option -2 = values returned in
format YYYYMMDD, type integer

  year   = floattointeger(JJAS(:,0))    ; Convert to integer for when using sprinti
  month  = floattointeger(JJAS(:,1))
  day    = floattointeger(JJAS(:,2))

  minimum!0    = "time"
  minimum&time = time
  maximum!0    = "time"
  maximum&time = time

  k=0                          ; First do loop cycles through times and picks out JJAS
dates
  j=0                          ; Time, date, max, &  min arrays are deleted then
reinitialized with a dimension size of 'j'
  t=0

  do t=0, num_time-1
   if ((utc_date(t,1)).eq.6) .or. ((utc_date(t,1)).eq.7) .or. ((utc_date(t,1)).eq.8) .or.
((utc_date(t,1)).eq.9) then
    delete(JJAS_time)
    delete(JJAS_date_array)
    delete(JJAS_max)
    delete(JJAS_min)
    k=t
    JJAS(k,:) = utc_date(t,:)
    j=j+1                                     ; j is a counter for JJAS dates
    JJAS_time       = new((/files,j/),float) ; reinitialize arrays with new dimension 'j'
    JJAS_date_array = new((/files,j/),integer)
    JJAS_max        = new((/files,j/),float)
    JJAS_min        = new((/files,j/),float)
   end if
  end do

  print("loop 1 complete")

  k=0                 ; 2nd do loop cycles back through times and fills JJAS time, date,
max, min arrays with appr values
  j=0
  t=0

  do t=0, num_time-1
   if ((utc_date(t,1)).eq.6) .or. ((utc_date(t,1)).eq.7) .or. ((utc_date(t,1)).eq.8) .or.
((utc_date(t,1)).eq.9) then
    JJAS_time(df,j) = time(t)
    JJAS_date_array(df,j) = JJAS_date(t)     ; fill arrays with JJAS time, date, max, min
values
    JJAS_max(df,j) = maximum(t)
    JJAS_min(df,j) = minimum(t)
    j=j+1
   end if
  end do

  tmin_avg(df) = avg(JJAS_min(df,:))  ; average of all JJAS min temps
  tmax_avg(df) = avg(JJAS_max(df,:))  ; average of all JJAS max temps

  print(j)
```

```
  print("loop 2 complete")

  doy = day_of_year(year, month, day)

  y=0                          ; Do Loop to rearrange JJAS times, dates, maxs into 2-D
arrays
  k=0
  do m=6,9
   if (m .eq. 6) .or. (m .eq. 9) then
    do d=1,30
     y=0
     do t=0,num_time-1
      if (month(t) .eq. m) .and. (day(t) .eq. d) then
       max_array(df,y,k) = maximum(t)
       min_array(df,y,k) = minimum(t)
       max_array_time(df,y,k) = time(t)
       max_array_date(df,y,k) = JJAS_date(t)
       y=y+1
      end if
     end do
    k=k+1
    end do
   else
    do d=1,31
     y=0
     do t=0,num_time-1
      if (month(t) .eq. m) .and. (day(t) .eq. d) then
       max_array(df,y,k) = maximum(t)
       min_array(df,y,k) = minimum(t)
       max_array_time(df,y,k) = time(t)
       max_array_date(df,y,k) = JJAS_date(t)
       y=y+1
      end if
     end do
    k=k+1
    end do
   end if
  end do

  print("Max_array done")

  do y=0,yrs-1         ; Do loop to indicate anomalies over 100
   do k=0,dys-1
    if (max_array(df,y,k) .ge. 100.) then
     over_max_array(df,y,k) = max_array(df,y,k) - 100.
    else
     over_max_array(df,y,k) = 0
    end if
   end do
  end do
  print("Over_max_array computed")

  do i=0,dys-1         ; Do loop to find the Long Term Daily Means
   LTDM(df,i) = avg(max_array(df,:,i))
  end do

  do y=0,yrs-1         ; Do Loop to find the daily anomalies
   do k=0,dys-1
    anomaly(df,y,k) = max_array(df,y,k) - LTDM(df,k)        ; Anomaly = Observed - LTDM
   end do
  end do
  print("Anomaly done")

  JJAS_SD(df) = stddev(max_array(df,:,:))    ; Find JJAS standard deviation for each city

  do y=0,yrs-1         ; Do Loop to compute normalized anomalies (anom/SD)
   do k=0,dys-1
    norm_anom(df,y,k) = anomaly(df,y,k) / JJAS_SD(df)
   end do
  end do
```

```
  amin = 10                          ; Minimum anomaly threshold
  amax = 15                          ; Maximum anomaly threshold

  N = num(ismissing(anomaly))        ; Number of anomaly missing values
  print(N)

  a=0                                       ; Counter for number of events that exceed
maximum anomaly threshold
  b=0                                       ; Counter for number of events that do NOT
exceed maximum anomaly threshold
  do y=0,yrs-1                              ; Do Loop to capture anomalies that exceed
the Maximum anomaly threshold
   do k=0,dys-1
    if (anomaly(df,y,k) .ge. amin) then
     over_anom(df,y,k) = anomaly(df,y,k)
     over_anom_date(df,y,k) = max_array_date(df,y,k)
     a=a+1
    else
     over_anom(df,y,k) = 0
     over_anom_date(df,y,k) = max_array_date(df,y,k)
     b=b+1
    end if
   end do
  end do

  print("Number of Anomalous Dates ="+a)
  print("Number of Non-Anomalous Dates ="+b)
  print("Over_anoms done")

  s=0          ; Do Loop that identifies event starting/ending dates
  d=0          ; Event defined as 3 or more consecutive days with all anoms greater than
or equal to  min anom threshold
               ; and at least 1 day greater than or equal to max anom threshold
  do y=0,yrs-1
   do k=0,dys-1
    if (k .eq. 0) then
     if (over_anom(df,y,k) .ne. 0) .and. (over_anom(df,y,k+1) .ne. 0) .and.
(over_anom(df,y,k+2) .ne. 0) then
      start_date(df,y,k) = over_anom_date(df,y,k)
      s=s+1
     else
      start_date(df,y,k) = 0
     end if
    end if

    if (k .ge. 1) .and. (k .le. dys-3) then
     if (over_anom(df,y,k).ne.0) .and. (over_anom(df,y,k-1).eq.0) .and.
(over_anom(df,y,k+1).ne.0) .and. (over_anom(df,y,k+2).ne.0) then
      start_date(df,y,k) = over_anom_date(df,y,k)
      s=s+1
     else
      start_date(df,y,k) = 0
     end if
    end if

    if (k .eq. dys-2) .or. (k .eq. dys-1) then
     start_date(df,y,k) = 0
    end if

    if (k .eq. 0) .or. (k .eq. 1) then
     end_date(df,y,k) = 0
    end if

    if (k .ge. 2) .and. (k .le. dys-2) then
     if (over_anom(df,y,k).ne.0) .and. (over_anom(df,y,k+1).eq.0) .and.
(over_anom(df,y,k-1).ne.0) .and. (over_anom(df,y,k-2).ne.0) then
      end_date(df,y,k) = over_anom_date(df,y,k)
      d=d+1
     else
      end_date(df,y,k) = 0
     end if
```

```
    end if

    if (k .eq. dys-1) then
     if (over_anom(df,y,k) .ne. 0) .and. (over_anom(df,y,k-1) .ne. 0) .and.
(over_anom(df,y,k-2) .ne. 0) then
      end_date(df,y,k) = over_anom_date(df,y,k)
      d=d+1
     else
      end_date(df,y,k) = 0
     end if
    end if

   end do
  end do

  print("Start/End Dates done")
  print("Number of Starting Dates = "+s)
  print("Number of Ending Dates = "+d)
                              ; Do Loop to find Tmax sums, Tmax averages, and # of days
of each event
  do y=0,yrs-1
   do k=0,dys-1
    if (start_date(df,y,k) .ne. 0) then
     j=1
     sum_max(df,y,k) = max_array(df,y,k)
     do i=1,j
      if (end_date(df,y,k+i) .eq. 0) then
       sum_max(df,y,k) = max_array(df,y,k+i) + sum_max(df,y,k)
       j=j+1
      else
       sum_max(df,y,k)  = sum_max(df,y,k) + max_array(df,y,k+i)
       ave_max(df,y,k)  = ((sum_max(df,y,k))/(i+1))
       num_days(df,y,k) = i+1
      end if
     end do
    else
     sum_max(df,y,k)  = 0
     ave_max(df,y,k)  = 0
     num_days(df,y,k) = 0
    end if
   end do
  end do

  print("Sum,Ave,Num done")
                       ; Do Loop to identify(0/1) event dates that have Tmaxs > the max
                       ; anom threshold.  Sum of the indicators tells how many above
threshold
                       ; Tmaxs each event has.  Only events with at least 1 Tmax > amax
are kept.
  do y=0,yrs-1
   do k=0,dys-1
    if (start_date(df,y,k) .ne. 0) then
     do i=0,num_days(df,y,k)-1
      if (over_anom(df,y,k+i) .lt. amax) then
       indicator(df,y,k+i) = 0
      else
       indicator(df,y,k+i) = 1
      end if
      if (i .eq. 0) then
       sum_test(df,y,k+i) = indicator(df,y,k+i)
      end if
      if (i .ne. 0) then
       sum_test(df,y,k+i) = indicator(df,y,k+i) + sum_test(df,y,k+i-1)
      end if
     end do
     sum_test(df,y,k) = sum_test(df,y,k+(num_days(df,y,k)-1))
    else
     sum_test(df,y,k) = 0
    end if
                       ;******** requires average max temp of event to be >= 100, or
NOT*******
```

```
      ;*************************************************************************

    if (sum_test(df,y,k) .eq. 0) .or. (ave_max(df,y,k) .lt. 100) then
     event_start(df,y,k) = 0
    else
     event_start(df,y,k) = start_date(df,y,k)
    end if

   end do
  end do

  print("Sum_Test done")
  print("Sum/Ave Maxs done")

  do y=0,yrs-1        ; Do Loop to rename the number of event days for each event
   do k=0,dys-1
    if (event_start(df,y,k) .ne. 0) then
     num_days_events(df,y,k) = num_days(df,y,k)
    else
     num_days_events(df,y,k) = 0
    end if
   end do
  end do

  days(df) = sum(num_days_events(df,:,:))           ; Number of events for each city
  print(days(df))

  print("Event,max,min arrays done")

  do y=0,yrs-1        ; Do Loop to find the max temp of events
   do k=0,dys-1
    if (event_start(df,y,k) .ne. 0) then
     temps = new((/num_days_events(df,y,k)/),float)
      do i=0,num_days_events(df,y,k)-1
       norm_anom_events(df,y,k+i) = norm_anom(df,y,k+i)
       temps(i) = max_array(df,y,k+i)
      end do
      print(temps)
      max_temp(df,y,k) = max(temps)
      print(max_temp(df,y,k))
      delete(temps)
     else
      max_temp(df,y,k) = 0
    end if
   end do
  end do
  norm_anom_avg(df) = avg(norm_anom_events(df,:,:))     ; avg of event normalized
anomalies
  norm_anom_avg_all(df) = avg(norm_anom(df,:,:))    ; avg of all normalized anomalies

  N=3                 ; Do Loop to compute 3-day anomaly averages and find the highest
one of the event
  do y=0,yrs-1
   do k=0,dys-1
    if (event_start(df,y,k) .ne. 0) then
     anom_ave = new((/num_days_events(df,y,k)-(N-1)/),float)
     anom_ave_date = new((/num_days_events(df,y,k)-(N-1)/),integer)
     anom_ave_perm = new((/num_days_events(df,y,k)-(N-1)/),integer)
     do i=0,num_days_events(df,y,k)-N
      anom_ave(i) = avg(anomaly(df,y,k+i:k+i+2))
;),anomaly(df,y,k+i+1),anomaly(df,y,k+i+2))
      anom_ave_date(i) = max_array_date(df,y,k+i)
     end do
     print(anom_ave)
     print(anom_ave_date)
     max_anom_ave(df,y,k) = max(anom_ave)
     anom_ave_perm = dim_pqsort(anom_ave, -1)
     max_anom_ave_date(df,y,k) = anom_ave_date(anom_ave_perm(0))
     print(max_anom_ave(df,y,k))
     print(max_anom_ave_date(df,y,k))
```

```
     delete(anom_ave)
     delete(anom_ave_date)
     delete(anom_ave_perm)
    else
     max_anom_ave(df,y,k) = 0
     max_anom_ave_date(df,y,k) = 0
    end if
   end do
  end do

  time_JJAS = new((/JJAS_days/),integer)     ; Do Loop to create timeline for following
plots
  do i=0,JJAS_days-1
   time_JJAS(i) = i
  end do

  time_anomaly = new((/dys/),integer)        ; Do Loop to create timeline for following
plots
  do i=0,dys-1
   time_anomaly(i) = i
  end do

;*************************************************************
; The code below creates an XY plot of the JJAS LTDMs.
;*************************************************************

  LTDM_plot = new((/dys/),float)

  LTDM_plot(:) = LTDM(df,:)

  wks1 = gsn_open_wks ("ps","IPM_LTDM_Plot_"+datafiles(df))

  res1                   = True                      ; plot mods desired
  res1@tiMainString      = datafiles(df)+" JJAS LTDMs"        ; add title

  res1@tiXAxisString     = "Date"                         ; xaxis string
  res1@tiYAxisString     = "LTDM"                 ; yaxis string

  res1@xyLineThicknesses = (/1.0,2.0/)                    ; make 2nd lines thicker
  res1@xyDashPattern = 0                                  ; Make curves all solid

  res1@xyLineColors      = (/"red","blue"/)               ; change line color
  res1@tmXBPrecision = 4                                  ; set the precision to 4
significant digits
  res1@trXMinF = 1
  res1@trXMaxF = 123

  res1@tmXBMode = "Explicit"
  res1@tmXBValues = (/0,14,30,44,61,75,92,106/)
  res1@tmXBLabels = (/"1 June","15 June","1 July","15 July","1 Aug","15 Aug","1 Sept","15
Sept"/)
  res1@tmXBMinorValues = ispan(0,121,1)
  res1@tmXBLabelFontHeightF = 0.01

  plot1  = gsn_csm_xy (wks1,ispan(0,dys-1,1),LTDM_plot,res1)             ; create plot
  print("JJAS LTDMs")

;*************************************************************
; The code below creates an XY plot of the JJAS anomalies.
;*************************************************************

  do i=0,yrs-1

  res2                   = True                      ; plot mods desired
  res2@tiMainString      = datafiles(df)+" JJAS Anomalies, Year " + sprinti("%0.4i",
i+1979)        ; add title

  res2@tiXAxisString     = "Date"                         ; xaxis string
  res2@tiYAxisString     = "Temperature Anomaly"                 ; yaxis string

  res2@xyLineThicknesses = (/1.0,2.0/)                    ; make 2nd lines thicker
```

```
  res2@xyDashPattern = 0                                  ; Make curves all solid

  res2@xyLineColors      = (/"red","blue"/)               ; change line color
  res2@tmXBPrecision = 4                                  ; set the precision to 4
significant digits
  res2@trXMinF = 1
  res2@trXMaxF = dys+1
                                                        ; create a reference line and shade
values above and below with
  res2@tmXBMode = "Explicit"
  res2@tmXBValues = (/0,14,30,44,61,75,92,106/)
  res2@tmXBLabels = (/"1 June","15 June","1 July","15 July","1 Aug","15 Aug","1 Sept","15
Sept"/)
  res2@tmXBMinorValues = ispan(0,121,1)
  res2@tmXBLabelFontHeightF = 0.01

                                                        ; selected colors. This is shading
array dsoid.
  res2@gsnYRefLine           = 0.0                             ; create a reference line
  res2@gsnYRefLineThicknessF = 2.0
; res2@gsnAboveYRefLineColor = (/"red"/)
; res2@gsnBelowYRefLineColor = (/"blue"/)
;*********************************
;
; polyline parameters used on both plots
;*********************************
; polyres2                  = True
; polyres2@gsLineThicknessF = 3.0
 ;*********************************
; res2@gsnAboveYRefLineColor = (/"red"/)                  ; above ref line fill red
; res2@gsnBelowYRefLineColor = (/"blue"/)                 ; below ref line fill blue
; (taken out so other plots could be plotted)  plot2  = gsn_csm_xy(wks2,ispan(0,dys-
1,1),anomaly(df,i,:),res2) ; create plot
; gsn_polyline(wks2,plot2,time_anomaly,(/anomaly_plot/),polyres2)   ; add polyline
; frame(wks)
  end do
;  print("Anomaly")


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

 asciiwrite(datafiles(df)+"_JJAS_max", JJAS_max(df,:))
 print("asciiwrite JJAS_max done")

  opt1       = True           ; File outs arrays to separate files, to be called in from
other programs
  opt1@title = "JJAS Tmaxs"
  opt1@fout  = datafiles(df)+"_max_array"
  write_matrix(max_array(df,:,:), "I3", opt1)

  print("opt1 fout done")

  opt2       = True
  opt2@title = "JJAS Tmax dates"
  opt2@fout  = datafiles(df)+"_max_array_date"
  write_matrix(max_array_date(df,:,:), "I8.8", opt2)

  print("opt2 fout done")

  opt3       = True
  opt3@title = "Anomalies"
  opt3@fout  = datafiles(df)+"_Anomalies"
  write_matrix(anomaly(df,:,:), "f6.2", opt3)

  print("opt3 fout done")

  opt4       = True
  opt4@title = "Average Tmax"
  opt4@fout  = datafiles(df)+"_ave_max"
  write_matrix(ave_max(df,:,:), "f6.2", opt4)

  print("opt4 fout done")
```

```
  opt5       = True
  opt5@title = "Number of Event Days"
  opt5@fout  = datafiles(df)+"_num_days"
  write_matrix(num_days_events(df,:,:), "I3", opt5)

  print("opt5 fout done")

  opt6       = True
  opt6@title = "Number of Event Days over Maximum Threshold"
  opt6@fout  = datafiles(df)+"_sum_test"
  write_matrix(sum_test(df,:,:), "f2.0", opt6)

  print("opt6 fout done")

  opt7       = True
  opt7@title = "Event Start Date"
  opt7@fout  = datafiles(df)+"_event_start"
  write_matrix(event_start(df,:,:), "I8.8", opt7)

  print("opt7 fout done")

  opt8       = True
  opt8@title = "Event End Date"
  opt8@fout  = datafiles(df)+"_end_date"
  write_matrix(end_date(df,:,:), "I8.8", opt8)

  print("opt8 fout done")

  print(datafiles(df)+" complete")

  if (df .eq. files-1) then
   corr_temp = new((/files/),float)
   do df=0,files-1
    corr_temp(df) = escorc(JJAS_max(3,:),JJAS_max(df,:))
   end do
   print(corr_temp(:))
  end if

  delete(time)
  delete(maximum)
  delete(minimum)

 end do                              ; The End Do of all End Dos

;*************************************************************************************
*************************
;*************************************************************************************
*************************
 print(anomaly)
 print(JJAS_SD)
 print(norm_anom)
 print(norm_anom_events)
 print(norm_anom_avg)
 print(norm_anom_avg_all)
 print(max_array)
 print(max_array_date)
 print(max_anom_ave)

 asciiwrite("JJAS_SD", JJAS_SD(:))
 asciiwrite("norm_anom_avg", norm_anom_avg(:))

 anomave    = new((/files,JJAS_days/),float)
 anomavedate = new((/files,JJAS_days/),integer)
 normanom   = new((/files,JJAS_days/),float)
 anomdate   = new((/files,JJAS_days/),integer)
 max_temp_2 = new((/files,JJAS_days/),float)
 maxarray   = new((/files,JJAS_days/),float)

 do df=0,files-1      ; transform 3-D arrays to 2-D arrays
  anomave(df,:) = ndtooned(max_anom_ave(df,:,:))
```

```
 anomavedate(df,:) = ndtooned(max_anom_ave_date(df,:,:))
 normanom(df,:) = ndtooned(norm_anom(df,:,:))
 anomdate(df,:) = ndtooned(max_array_date(df,:,:))
 max_temp_2(df,:) = ndtooned(max_temp(df,:,:))
 maxarray(df,:) = ndtooned(max_array(df,:,:))

 print(max_temp_2(df,:))

end do

ave_max_long        = new((/files,JJAS_days/),float)
ave_max_long_date   = new((/files,JJAS_days/),integer)
anom_long          = new((/files,JJAS_days/),float)
ave_max_sort        = new((/files,JJAS_days/),integer)
ave_max_sorted      = new((/files,JJAS_days/),float)
ave_max_sorted_date = new((/files,JJAS_days/),integer)
max_long            = new((/files,JJAS_days/),float)
max_long_date       = new((/files,JJAS_days/),integer)
num_days_long       = new((/files,JJAS_days/),integer)
date_num            = new((/files/),integer)

a=0   ; transform 3-D arrays to 2-D arrays
b=0
d=0
do df=0,files-1
 a=0
 b=0
 d=0
 do y=0,yrs-1
  do k=0,dys-1
   if (event_start(df,y,k) .ne. 0) then
    ave_max_long(df,a) = ave_max(df,y,k)
    ave_max_long_date(df,a) = max_array_date(df,y,k)
    num_days_long(df,a) = num_days_events(df,y,k)
    nd=num_days_events(df,y,k)
    do n=0,nd-1
     max_long(df,b+n) = max_array(df,y,k+n)
     max_long_date(df,b+n) = max_array_date(df,y,k+n)
     anom_long(df,b+n) = anomaly(df,y,k+n)
     d=d+1
    end do
    a=a+1
    b=b+nd
   else
    ave_max_long(df,a) = 0
    ave_max_long_date(df,a) = 0
    num_days_long(df,a) = 0
    a=a+1
   end if
  end do
 end do
 date_num(df) = d
end do

ave_max_date         = new((/files,JJAS_days/),integer)
anomave_perm         = new((/files,JJAS_days/),integer)
max_temp_2_perm      = new((/files,JJAS_days/),integer)
anomave_sorted       = new((/files,JJAS_days/),float)
anomavedate_sorted   = new((/files,JJAS_days/),integer)
anomdate_sorted      = new((/files,JJAS_days/),integer)
max_temp_date_sorted = new((/files,JJAS_days/),integer)
max_temp_2_sorted = new((/files,JJAS_days/),float)

                     ; reorder arrays in descending order to find
                     ; the max values and their dates

do df=0,files-1
 ave_max_sort(df,:)   = dim_pqsort(ave_max_long(df,:),-1)
 anomave_perm(df,:)   = dim_pqsort(anomave(df,:),-1)
 max_temp_2_perm(df,:) = dim_pqsort(max_temp_2(df,:),-1)
end do
```

```
 print(max_temp_2_perm)

 do df=0,files-1
  do i=0,JJAS_days-1
   anomave_sorted(df,i) = anomave(df,anomave_perm(df,i))
   anomavedate_sorted(df,i) = anomavedate(df,anomave_perm(df,i))
   anomdate_sorted(df,i) = anomdate(df,anomave_perm(df,i))
   max_temp_2_sorted(df,i) = max_temp_2(df,max_temp_2_perm(df,i))
   max_temp_date_sorted(df,i) = anomdate(df,max_temp_2_perm(df,i))
  end do
 end do
 print(max_temp_2_sorted)
 print(max_temp_date_sorted)

;***************
; Print outs for the different Ranking Methods
;***************

 rank_end_date_try = new((/files,JJAS_days/),integer)
 num_days_rank_try = new((/files,JJAS_days/),integer)

 do df=0,files-1
  r=0
  print("HW Ranking by Highest Onset Maximum Temperature")
  print("City = "+datafiles(df))
  print("Ranking StartDate   EndDate  TMax  NumDays")
  do j=0,JJAS_days-1
   if (max_temp_2_sorted(df,j) .ne. 0) then
    r=r+1
    do y=0,yrs-1
     do k=0,dys-1
      if (max_temp_date_sorted(df,j) .eq. event_start(df,y,k)) then
       rank_end_date_try(df,j) = end_date(df,y,k+(num_days(df,y,k)-1))
       num_days_rank_try(df,j) = num_days_events(df,y,k)
      end if
     end do
    end do
    print("  "+r+"    "+sprinti("%8.0i", max_temp_date_sorted(df,j))+"
"+sprinti("%8.0i", rank_end_date_try(df,j))+"   " \
          +sprintf("%6.2f", max_temp_2_sorted(df,j))+"      "+sprinti("%2.0",
num_days_rank_try(df,j)))
   end if
  end do
 end do


 rank_end_date_1 = new((/files,JJAS_days/),integer)
 num_days_rank_1 = new((/files,JJAS_days/),integer)

 do df=0,files-1
  r=0
  print("HW Ranking by Highest "+N+"-day Average Maximum Temperature Anomaly")
  print("City = "+datafiles(df))
  print("Ranking StartDate   EndDate  AnomStartDate Ave TMax Anom  NumDays")
  do j=0,JJAS_days-1
   if (anomave_sorted(df,j) .ne. 0) then
    r=r+1
    do y=0,yrs-1
     do k=0,dys-1
      if (anomdate_sorted(df,j) .eq. event_start(df,y,k)) then
       rank_end_date_1(df,j) = end_date(df,y,k+(num_days(df,y,k)-1))
       num_days_rank_1(df,j) = num_days_events(df,y,k)
      end if
     end do
    end do
    print("  "+r+"    "+sprinti("%8.0i", anomdate_sorted(df,j))+"    "+sprinti("%8.0i",
rank_end_date_1(df,j))+"   " \
          +sprinti("%8.0i", anomavedate_sorted(df,j))+"      "+sprintf("%6.2f",
anomave_sorted(df,j))+"      " \
          +sprinti("%2.0", num_days_rank_1(df,j)))
   end if
```

```
 end do
end do

top_rank = 15
do df=0,files-1
 r=0
 print(datafiles(df)+" Top "+top_rank+" Events")
 print("Ranked by Highest "+N+"-day Average Maximum Temperature Anomaly")
 print("Ranking StartDate   EndDate  AnomStartDate Tmax Average")
 do j=0,top_rank-1                    ; Selecting the top top_rank events
  r=r+1
  if (anomave_sorted(df,j) .ne. 0) then
   print("  "+r+"  "+sprinti("%8.0i", anomdate_sorted(df,j))+"   "+sprinti("%8.0i",
rank_end_date_1(df,j))+"   " \
         +sprinti("%8.0i", anomavedate_sorted(df,j))+"       "+sprintf("%6.2f",
anomave_sorted(df,j))+"      " \
         +sprinti("%2.0", num_days_rank_1(df,j)))
  end if
 end do
end do

Sac_norm_anoms = new((/files,top_rank/),float)
Sac_norm_anoms_avg = new((/files/),float)
Sacdate_target_3day = new((/files,top_rank/),float)
Sacdate_target_3day_avg = new((/files/),float)

do df=0,files-1      ; Find normalized anomalies on Sac Event dates
 do j=0,top_rank-1   ; at all stations
  do k=0,JJAS_days-1
   if (anomavedate_sorted(3,j) .ne. 0) then
    if (anomavedate_sorted(3,j) .eq. anomdate(df,k)) then
     Sac_norm_anoms(df,j) = normanom(df,k)
    end if
   end if
  end do
 end do
 Sac_norm_anoms_avg(df) = avg(Sac_norm_anoms(df,:))
end do
print(Sac_norm_anoms)
print(Sac_norm_anoms_avg)

do df=0,files-1
 do j=0,top_rank-1
  do k=0,JJAS_days-1
   if (anomdate_sorted(3,j) .ne. 0) then
    if (anomdate_sorted(3,j) .eq. anomdate(df,k)) then
     Sacdate_target_3day(df,j) = maxarray(df,k)
    end if
   end if
  end do
 end do
 Sacdate_target_3day_avg(df) = avg(Sacdate_target_3day(df,:))
end do
print(Sacdate_target_3day)
print(Sacdate_target_3day_avg)

asciiwrite("Sac_norm_anoms_avg", Sac_norm_anoms_avg(:))
asciiwrite("Sacdate_target_3day_avg", Sacdate_target_3day_avg(:))

do df=0,files-1
 qsort(ave_max_long(df,:))
 ave_max_sorted(df,:) = ave_max_long(df,::-1)
end do

do df=0,files-1
 do j=0,JJAS_days-1
  ave_max_date(df,j) = ave_max_long_date(df,ave_max_sort(df,j))
 end do
end do

ave_max_rank = new((/files,JJAS_days/),integer)
```

```
 b=0
 do df=0,files-1
  b=0
  do j=0,JJAS_days-1
   if (ave_max_sorted(df,j) .ne. 0) then
    ave_max_rank(df,j) = b+1
    b=b+1
   end if
  end do
 end do

;*************
; Print outs of more Ranking Methods
;************

 rank_end_date = new((/files,JJAS_days/),integer)
 num_days_rank = new((/files,JJAS_days/),integer)
 do df=0,files-1
  print("HW Ranking by Highest Average Maximum Temperature")
  print("City = "+datafiles(df))
  print("Ranking StartDate   EndDate  Tmax Average")
  do j=0,JJAS_days-1
   if (ave_max_sorted(df,j) .ne. 0) then
    do y=0,yrs-1
     do k=0,dys-1
      if (ave_max_date(df,j) .eq. event_start(df,y,k)) then
       rank_end_date(df,j) = end_date(df,y,k+(num_days(df,y,k)-1))
       num_days_rank(df,j) = num_days_events(df,y,k)
      end if
     end do
    end do
    print("  "+sprinti("%2.0i", ave_max_rank(df,j))+"  "+sprinti("%8.0i",
ave_max_date(df,j))+"   "+sprinti("%8.0i", rank_end_date(df,j))+"   "+sprintf("%6.2f",
ave_max_sorted(df,j)))
   end if
  end do
 end do

 top_rank = 15
 do df=0,files-1
  print(datafiles(df)+" Top "+top_rank+" Events")
  print("Ranked by Highest Average Maximum Temperature")
  print("Ranking StartDate   EndDate  Tmax Average")
  do j=0,top_rank-1                  ; Selecting the top top_rank events
   if (ave_max_sorted(df,j) .ne. 0) then
    print("  "+sprinti("%2.0i", ave_max_rank(df,j))+"  "+sprinti("%8.0i",
ave_max_date(df,j))+"   "+sprinti("%8.0i", rank_end_date(df,j))+"   "+sprintf("%6.2f",
ave_max_sorted(df,j)))
   end if
  end do
 end do

 top_rank_ave      = new((/files,top_rank/),float)
 top_rank_ave_date = new((/files,top_rank/),integer)
 top_rank_anomave  = new((/files,top_rank/),float)
 top_rank_anomdate = new((/files,top_rank/),integer)
 top_rank_max      = new((/files,top_rank/),float)
 top_rank_max_date = new((/files,top_rank/),integer)
 top_temps         = new((/files,JJAS_days/),float)
 top_temps_N       = new((/files,JJAS_days/),float)
 top_temps_max     = new((/files,JJAS_days/),float)
 top_dates         = new((/files,JJAS_days/),integer)
 top_dates_N       = new((/files,JJAS_days/),integer)
 top_dates_max     = new((/files,JJAS_days/),integer)
 top_anoms         = new((/files,JJAS_days/),float)
 top_anoms_N       = new((/files,JJAS_days/),float)
 top_anoms_max     = new((/files,JJAS_days/),float)
 count             = new((/files/),integer)
 count_Nday        = new((/files/),integer)
```

```
 counter=0
 counter_Nday=0
 do df=0,files-1
  counter=0
  counter_Nday=0
  do j=0,JJAS_days-1
   if (ave_max_sorted(df,j) .ne. 0) then
    counter=counter+1
   end if
   if (anomave_sorted(df,j) .ne. 0) then
    counter_Nday=counter_Nday+1
   end if
  end do
  count(df)  = counter
  count_Nday(df) = counter_Nday
 end do
 print(count)
 print(count_Nday)

 l=0           ; grab top rank dates of certain max values
 i=0
 b=0
 c=0
 do df=0,files-1
  i=0
  b=0
  c=0
  if (count(df) .le. top_rank) then
   do j=0,count(df)-1
    if (ave_max_sorted(df,j) .ne. 0) then
     top_rank_ave(df,i) = ave_max_sorted(df,j)                    ; ONLY top N events,
no holes
     top_rank_ave_date(df,i) = ave_max_date(df,j)
     i=i+1
    end if
    if (anomave_sorted(df,j) .ne. 0) then
     top_rank_anomave(df,b) = anomave_sorted(df,j)
     top_rank_anomdate(df,b) = anomdate_sorted(df,j)
     b=b+1
    end if
    if (max_temp_2_sorted(df,j) .ne. 0) then
     top_rank_max(df,c) = max_temp_2_sorted(df,j)
     top_rank_max_date(df,c) = max_temp_date_sorted(df,j)
     c=c+1
    end if
   end do
  end if

  i=0
  b=0
  c=0
  if (count(df) .gt. top_rank) then
   do j=0,top_rank-1
    if (ave_max_sorted(df,j) .ne. 0) then
     top_rank_ave(df,i) = ave_max_sorted(df,j)
     top_rank_ave_date(df,i) = ave_max_date(df,j)
     i=i+1
    end if
    if (anomave_sorted(df,j) .ne. 0) then
     top_rank_anomave(df,b) = anomave_sorted(df,j)
     top_rank_anomdate(df,b) = anomdate_sorted(df,j)
     b=b+1
    end if
    if (max_temp_2_sorted(df,j) .ne. 0) then
     top_rank_max(df,c) = max_temp_2_sorted(df,j)
     top_rank_max_date(df,c) = max_temp_date_sorted(df,j)
     c=c+1
    end if
   end do
  end if
 end do
```

```
top_rank_onset      = new((/files,top_rank/),float)
top_rank_onset_Nday = new((/files,top_rank/),float)
top_rank_onset_max  = new((/files,top_rank/),float)

do df=0,files-1      ; Do Loop to grab the top rank onset dates
 if (count(df) .le. top_rank) then
  do k=0,count(df)-1
   do j=0,date_num(df)-1
    if (top_rank_ave_date(df,k) .eq. max_long_date(df,j)) then
     top_rank_onset(df,k) = max_long(df,j)
    end if
    if (top_rank_anomdate(df,k) .eq. max_long_date(df,j)) then
     top_rank_onset_Nday(df,k) = max_long(df,j)
    end if
    if (top_rank_max_date(df,k) .eq. max_long_date(df,j)) then
     top_rank_onset_max(df,k) = max_long(df,j)
    end if
   end do
  end do
 end if

 if (count(df) .gt. top_rank) then
  do k=0,top_rank-1
   do j=0,date_num(df)-1
    if (top_rank_ave_date(df,k) .eq. max_long_date(df,j)) then
     top_rank_onset(df,k) = max_long(df,j)
    end if
    if (top_rank_anomdate(df,k) .eq. max_long_date(df,j)) then
     top_rank_onset_Nday(df,k) = max_long(df,j)
    end if
    if (top_rank_max_date(df,k) .eq. max_long_date(df,j)) then
     top_rank_onset_max(df,k) = max_long(df,j)
    end if
   end do
  end do
 end if
end do
print(top_rank_onset_max)

target_ave      = new((/files/),float)
target_ave_Nday = new((/files/),float)
target_ave_max  = new((/files/),float)
do df=0,files-1
 target_ave(df)      = avg(top_rank_onset(df,:))
 target_ave_Nday(df) = avg(top_rank_onset_Nday(df,:))
 target_ave_max(df)  = avg(top_rank_onset_max(df,:))
end do

do df=0,files-1
 asciiwrite(datafiles(df)+"_top_rank_onset", top_rank_onset(df,:))
 asciiwrite(datafiles(df)+"_top_rank_onset_Nday", top_rank_onset_Nday(df,:))
 asciiwrite(datafiles(df)+"_top_rank_onset_max", top_rank_onset_max(df,:))
end do
print("asciiwrite top_rank_onsets done")


asciiwrite("target_ave", target_ave(:))
asciiwrite("target_ave_Nday", target_ave_Nday(:))
asciiwrite("target_ave_max", target_ave_max(:))
print("asciiwrite target_aves done")

g=0                                   ; Do Loop to create an array of all events
b=0
f=0
j=0
i=0
jmax=0
do df=0,files-1
 b=0
 do y=0,yrs-1
```

```
   g=0
   do k=0,dys-1
    if (event_start(df,y,k) .ne. 0) then
     j=num_days_events(df,y,k)
     i=i+1                              ; i/ counter for number of events in ALL cities,
ALL years (grand total)
     b=b+1                              ; b/ counter for number of events in each city
file, ALL years
     g=g+1                              ; g/ counter for number of events EACH YEAR in
each city file
     if (j .ge. jmax) then
      jmax = j
     else
      jmax = jmax
     end if
    end if
   end do
   g_total(df,y) = g
  end do
  b_max(df) = b
  print(datafiles(df)+" Number of Events = "+b_max(df))
  j_max(df) = jmax
 end do
 print("Grand Total # of Events, ALL cities, ALL years = "+i)

 total_b = sum(b_max)
 bmax  = max(b_max(:))
 jjmax = max(j_max(:))
 g_total_sum = sum(g_total)

 tmin_year_avg = new((/files,yrs/),float)       ; 2-D array of Yearly Minimum Temp
averages
 tmax_year_avg = new((/files,yrs/),float)
 do df=0,files-1                                ; Do Loop to compute Minimum Temp
average; each year, for every city
  do y=0,yrs-1
   tmin_year_avg(df,y) = avg(min_array(df,y,:))
   tmax_year_avg(df,y) = avg(max_array(df,y,:))
  end do
 end do

 print(tmin_year_avg)
 print(tmax_year_avg)
 print(tmin_avg)
 print(tmax_avg)

 total_days = sum(days)         ; Number of total events, all years, all cities
 print(days)
 print(total_days)
 num_days_sum = sum(num_days_events)
 print(num_days_sum)

 event_array           = new((/files,bmax,jjmax/),float)
 event_dates_array     = new((/files,bmax,jjmax/),integer)
 event_array_all       = new((/num_days_sum/),float)           ; 1-D array of all event
Tmaxs
 event_dates_array_all = new((/num_days_sum/),integer)           ; 1-D array of all
event dates

 b=0           ; Create array of event dates
 f=0
 j=0
 do df=0,files-1
  b=0
  do y=0,yrs-1
   do k=0,dys-1
    if (event_start(df,y,k) .ne. 0) then
     b=b+1                                                      ; counter for number of
ALL events
     j=num_days_events(df,y,k)
     do n=0,j-1
```

```
     event_array(df,b-1,n) = max_array(df,y,k+n)
     event_dates_array(df,b-1,n) = max_array_date(df,y,k+n)
     event_array_all(f+n) = event_array(df,b-1,n)
     event_dates_array_all(f+n) = event_dates_array(df,b-1,n)
    end do
    event_max(df,y,k) = max(event_array(df,b-1,:))
    event_min(df,y,k) = min(event_array(df,b-1,:))
    f=f+j
    print(f)
   else
    event_max(df,y,k) = 0
    event_min(df,y,k) = 0
   end if
  end do
 end do
end do
print(event_max)

eventmaxes = new((/files,JJAS_days/),float)
eventdates = new((/files,JJAS_days/),integer)

do df=0,files-1
  eventmaxes(df,:)  = ndtooned(event_max(df,:,:))
  eventdates(df,:)  = ndtooned(max_array_date(df,:,:))
end do

eventmaxes_perm = new((/files,JJAS_days/),integer)
eventmaxes_sorted = new((/files,JJAS_days/),float)
eventdates_sorted = new((/files,JJAS_days/),integer)

do df=0,files-1
  eventmaxes_perm(df,:) = dim_pqsort(eventmaxes(df,:),-1)
end do

do df=0,files-1
 do i=0,JJAS_days-1
   eventmaxes_sorted(df,i) = eventmaxes(df,eventmaxes_perm(df,i))
   eventdates_sorted(df,i) = eventdates(df,eventmaxes_perm(df,i))
 end do
end do

do df=0,files-1
 do j=0,JJAS_days-1
  top_temps(df,j) = 0
  top_temps_N(df,j) = 0
  top_temps_max(df,j) = 0
  top_dates(df,j) = 0
  top_dates_N(df,j) = 0
  top_dates_max(df,j) = 0
 end do
end do

rank_end_date_2 = new((/files,JJAS_days/),integer)
num_days_rank_2 = new((/files,JJAS_days/),integer)

do df=0,files-1
 do j=0,JJAS_days-1
  if (eventmaxes_sorted(df,j) .ne. 0) then
   do y=0,yrs-1
    do k=0,dys-1
     if (eventdates_sorted(df,j) .eq. event_start(df,y,k)) then
      rank_end_date_2(df,j) = end_date(df,y,k+(num_days(df,y,k)-1))
      num_days_rank_2(df,j) = num_days_events(df,y,k)
     end if
    end do
   end do
  end if
 end do
end do

b=0
```

```
do df=0,files-1
 b=0
 if (count(df) .lt. top_rank) then
  do k=0,count(df)-1
   do j=0,date_num(df)-1
    if (ave_max_date(df,k) .eq. max_long_date(df,j)) then
     nd=num_days_rank(df,k)
     do n=0,nd-1
      top_dates(df,b+n) = max_long_date(df,j+n)
      top_temps(df,b+n) = max_long(df,j+n)
      top_anoms(df,b+n) = anom_long(df,j+n)
     end do
     b=b+nd
    end if
   end do
  end do
 end if

 b=0
 if (count(df) .ge. top_rank) then
  do k=0,top_rank-1
   do j=0,date_num(df)-1
    if (ave_max_date(df,k) .eq. max_long_date(df,j)) then
     nd=num_days_rank(df,k)
     do n=0,nd-1
      top_dates(df,b+n) = max_long_date(df,j+n)
      top_temps(df,b+n) = max_long(df,j+n)
      top_anoms(df,b+n) = anom_long(df,j+n)
     end do
     b=b+nd
    end if
   end do
  end do
 end if
end do

b=0                   ; Do Loop to grab top rank dates, temps, and anoms based
do df=0,files-1       ; on Nday ranking
 b=0
 if (count(df) .lt. top_rank) then
  do k=0,count(df)-1
   do j=0,date_num(df)-1
    if (anomdate_sorted(df,k) .eq. max_long_date(df,j)) then
     nd=num_days_rank_1(df,k)
     do n=0,nd-1
      top_dates_N(df,b+n) = max_long_date(df,j+n)
      top_temps_N(df,b+n) = max_long(df,j+n)
      top_anoms_N(df,b+n) = anom_long(df,j+n)
     end do
     b=b+nd
    end if
   end do
  end do
 end if

 b=0
 if (count(df) .ge. top_rank) then
  do k=0,top_rank-1
   do j=0,date_num(df)-1
    if (anomdate_sorted(df,k) .eq. max_long_date(df,j)) then
     nd=num_days_rank_1(df,k)
     do n=0,nd-1
      top_dates_N(df,b+n) = max_long_date(df,j+n)
      top_temps_N(df,b+n) = max_long(df,j+n)
      top_anoms_N(df,b+n) = anom_long(df,j+n)
     end do
     b=b+nd
    end if
   end do
  end do
 end if
```

```
 end do


 b=0           ; Do Loop to grab top rank dates based on max ranking
 do df=0,files-1
  b=0
  if (count(df) .lt. top_rank) then
   do k=0,count(df)-1
    do j=0,date_num(df)-1
     if (max_temp_date_sorted(df,k) .eq. max_long_date(df,j)) then
      nd=num_days_rank_2(df,k)
      do n=0,nd-1
       top_dates_max(df,b+n) = max_long_date(df,j+n)
       top_temps_max(df,b+n) = max_long(df,j+n)
       top_anoms_max(df,b+n) = anom_long(df,j+n)
      end do
      b=b+nd
     end if
    end do
   end do
  end if

  b=0
  if (count(df) .ge. top_rank) then
   do k=0,top_rank-1
    do j=0,date_num(df)-1
     if (max_temp_date_sorted(df,k) .eq. max_long_date(df,j)) then
      nd=num_days_rank_2(df,k)
      do n=0,nd-1
       top_dates_max(df,b+n) = max_long_date(df,j+n)
       top_temps_max(df,b+n) = max_long(df,j+n)
       top_anoms_max(df,b+n) = anom_long(df,j+n)
      end do
      b=b+nd
     end if
    end do
   end do
  end if
 end do

 do df=0,files-1
  asciiwrite(datafiles(df)+"_Top_Event_Dates", top_dates(df,:))
  asciiwrite(datafiles(df)+"_Top_Event_Dates_N", top_dates_N(df,:))
  asciiwrite(datafiles(df)+"_Top_Event_Dates_max", top_dates_max(df,:))
 end do

;***************
; Print outs of more rankings
;***************

 do df=0,files-1
  r=0
  print("HW Ranking by Highest Event Maximum Temperature")
  print("City = "+datafiles(df))
  print("Ranking StartDate   EndDate  Ave TMax Anom  NumDays")
  do j=0,JJAS_days-1
   if (eventmaxes_sorted(df,j) .ne. 0) then
    r=r+1
    print("  "+r+"    "+sprinti("%8.0i", eventdates_sorted(df,j))+"    "+sprinti("%8.0i",
rank_end_date_2(df,j))+"   " \
          +sprintf("%6.2f", eventmaxes_sorted(df,j))+"      "+sprinti("%2.0",
num_days_rank_2(df,j)))
   end if
  end do
 end do

 top_rank = 15
 do df=0,files-1
  r=0
  print(datafiles(df)+" Top "+top_rank+" Events")
  print("Ranked by Highest Event Maximum Temperature")
```

```
  print("Ranking StartDate   EndDate  Tmax Average")
  do j=0,top_rank-1                    ; Selecting the top top_rank events
   r=r+1
   if (eventmaxes_sorted(df,j) .ne. 0) then
    print("  "+r+"  "+sprinti("%8.0i", eventdates_sorted(df,j))+"   "+sprinti("%8.0i",
rank_end_date_2(df,j))+"   " \
          +sprintf("%6.2f", eventmaxes_sorted(df,j))+"      "+sprinti("%2.0",
num_days_rank_2(df,j)))
   end if
  end do
 end do

 Sacdate_target_max     = new((/files,top_rank/),float)
 Sacdate_target_max_avg = new((/files/),float)

 do df=0,files-1
  do j=0,top_rank-1
   do k=0,JJAS_days-1
    if (eventmaxes_sorted(3,j) .ne. 0) then
     if (eventdates_sorted(3,j) .eq. anomdate(df,k)) then
      Sacdate_target_max(df,j) = maxarray(df,k)
     end if
    end if
   end do
  end do
  Sacdate_target_max_avg(df) = avg(Sacdate_target_max(df,:))
 end do
 print(Sacdate_target_max)
 print(Sacdate_target_max_avg)

 asciiwrite("Sacdate_target_max_avg", Sacdate_target_max_avg(:))


 do df=0,files-1
  print(" ")
  print("City="+datafiles(df))
  print("Number of events = "+b_max(df))
  print("Min Anomaly Threshold="+amin)
  print("Max Anomaly Threshold="+amax)
                                          ; Do Loop to print out table of events
and event variables for each city
  i=0
  print("Event# EventStart  EndDate  MaxT MinT AvgTemp NumDays SumTest")
  do y=0,yrs-1
   do k=0,dys-1
    if (event_start(df,y,k) .ne. 0) then
     i=i+1
     print("  "+sprinti("%2.0i", i)+"    "+sprinti("%0.8i", event_start(df,y,k))+"  " \
           +sprinti("%0.8i", end_date(df,y,k+(num_days(df,y,k)-1)))+"
"+sprintf("%6.2f", event_max(df,y,k))+"  " \
           +sprintf("%6.2f", event_min(df,y,k))+"  "+sprintf("%6.2f", ave_max(df,y,k))+"
" \
           +sprinti("%4.0i", num_days(df,y,k))+"      "+sprintf("%2.0f",
sum_test(df,y,k)))
    end if
   end do
  end do
 end do

 do df=0,files-1
  print("City = "+datafiles(df))
  print("Year  # of Events")
  do y=0,yrs-1
   print(sprinti("%2.0i", y)+"       "+sprinti("%2.0i", g_total(df,y)))
  end do
 end do

 do df=0,files-1
  opt10       = True                          ; File outs arrays to separate files, to
be called in from other programs
  opt10@title = "JJAS Event Dates"
```

```
  opt10@fout  = datafiles(df)+"_event_dates_array"
  write_matrix(event_dates_array(df,:,:), "I8.8", opt10)
 end do

;*******************
; Create arrays of the top N event based on
; the different ranking methods
;*******************

 do df=0,files-1
  if (count(df) .le. top_rank) then

   if (df .eq. 0) then
    if (count(df) .ne. 0) then
    top_0 = new((/count(df)/),float)
    top_0_N = new((/count(df)/),float)
    do d=0,count(df)-1
     top_0(d) = ave_max_sorted(df,d)
     top_0_N(d) = anomave_sorted(df,d)
    end do
   end if
   end if

   if (df .eq. 1) then
    if (count(df) .ne. 0) then
    top_1 = new((/count(df)/),float)
    top_1_N = new((/count(df)/),float)
    do d=0,count(df)-1
     top_1(d) = ave_max_sorted(df,d)
     top_1_N(d) = anomave_sorted(df,d)
    end do
   end if
   end if

   if (df .eq. 2) then
    if (count(df) .ne. 0) then
    top_2 = new((/count(df)/),float)
    top_2_N = new((/count(df)/),float)
    do d=0,count(df)-1
     top_2(d) = ave_max_sorted(df,d)
     top_2_N(d) = anomave_sorted(df,d)
    end do
   end if
   end if

   if (df .eq. 3) then
    if (count(df) .ne. 0) then
    top_3 = new((/count(df)/),float)
    top_3_N = new((/count(df)/),float)
    do d=0,count(df)-1
     top_3(d) = ave_max_sorted(df,d)
     top_3_N(d) = anomave_sorted(df,d)
    end do
   end if
   end if
*
*
*  ...All cities computed in same way
*
*
   if (df .eq. 27) then
    if (count(df) .ne. 0) then
     top_27 = new((/count(df)/),float)
    top_27_N = new((/count(df)/),float)
     do d=0,count(df)-1
      top_27(d) = ave_max_sorted(df,d)
     top_27_N(d) = anomave_sorted(df,d)
     end do
    end if
   end if
```

```
   if (df .eq. 28) then
    if (count(df) .ne. 0) then
     top_28 = new((/count(df)/),float)
    top_28_N = new((/count(df)/),float)
     do d=0,count(df)-1
      top_28(d) = ave_max_sorted(df,d)
     top_28_N(d) = anomave_sorted(df,d)
     end do
    end if
   end if

   if (df .eq. 29) then
    if (count(df) .ne. 0) then
     top_29 = new((/count(df)/),float)
    top_29_N = new((/count(df)/),float)
     do d=0,count(df)-1
      top_29(d) = ave_max_sorted(df,d)
     top_29_N(d) = anomave_sorted(df,d)
     end do
    end if
   end if

  end if

  if (count(df) .gt. top_rank) then
   if (df .eq. 0) then
    top_0 = new((/top_rank/),float)
    top_0_N = new((/top_rank/),float)
    do d=0,top_rank-1
     top_0(d) = ave_max_sorted(df,d)
     top_0_N(d) = anomave_sorted(df,d)
    end do
   print(dimsizes(top_0))
   print(top_0)
   end if

   if (df .eq. 1) then
    top_1 = new((/top_rank/),float)
    top_1_N = new((/top_rank/),float)
    do d=0,top_rank-1
     top_1(d) = ave_max_sorted(df,d)
     top_1_N(d) = anomave_sorted(df,d)
    end do
   print(top_1)
   print(dimsizes(top_1))
   end if
*
* ...All done in same way...
*
   if (df .eq. 29) then
    top_29 = new((/top_rank/),float)
    top_29_N = new((/top_rank/),float)
    do d=0,top_rank-1
     top_29(d) = ave_max_sorted(df,d)
     top_29_N(d) = anomave_sorted(df,d)
    end do
   end if

  end if
 end do

;************************
; Compute correlations of top temperatures
;************************
'

 if (dimsizes(top_3) .ge. dimsizes(top_0)) .and. (dimsizes(top_0) .gt. 0) then
  delete(top_3)
  delete(top_3_N)
  top_3 = new((/dimsizes(top_0)/),float)
  top_3_N = new((/dimsizes(top_0)/),float)
  do d=0,dimsizes(top_0)-1
```

```
   top_3(d) = ave_max_sorted(3,d)
   top_3_N(d) = anomave_sorted(3,d)
  end do
  print(top_3)
  print(top_3_N)
 end if

 if (dimsizes(top_3) .lt. dimsizes(top_0)) .and. (dimsizes(top_0) .gt. 0) then
  delete(top_0)
  delete(top_0_N)
  top_0 = new((/dimsizes(top_3)/),float)
  top_0_N = new((/dimsizes(top_3)/),float)
  do d=0,dimsizes(top_3)-1
   top_0(d) = ave_max_sorted(0,d)
   top_0_N(d) = anomave_sorted(0,d)
  end do
  print(top_0)
  print(top_0_N)
 end if

 corr_top_0 = escorc(top_3(:),top_0(:))
 corr_top_0_N = escorc(top_3_N(:),top_0_N(:))
 print(corr_top_0)
 print(corr_top_0_N)
*
* ...All done in same way...
*
 if (dimsizes(top_3) .ge. count(29)) .and. (count(29) .gt. 0) then
  delete(top_3)
  delete(top_3_N)
  top_3 = new((/dimsizes(top_29)/),float)
  top_3_N = new((/dimsizes(top_29)/),float)
  do d=0,dimsizes(top_29)-1
   top_3(d) = ave_max_sorted(3,d)
   top_3_N(d) = anomave_sorted(3,d)
  end do
  print(top_29)
  print(top_29_N)
 corr_top_29 = escorc(top_3(:),top_29(:))
 corr_top_29_N = escorc(top_3_N(:),top_29_N(:))
 print(corr_top_29)
 print(corr_top_29_N)
 end if

 if (dimsizes(top_3) .lt. count(29)) .and. (count(29) .gt. 0) then
  delete(top_29)
  delete(top_29_N)
  top_29 = new((/dimsizes(top_3)/),float)
  top_29_N = new((/dimsizes(top_3)/),float)
  do d=0,dimsizes(top_3)-1
   top_29(d) = ave_max_sorted(29,d)
   top_29_N(d) = anomave_sorted(29,d)
  end do
  print(top_29)
  print(top_29_N)
 corr_top_29 = escorc(top_3(:),top_29(:))
 corr_top_29_N = escorc(top_3_N(:),top_29_N(:))
 print(corr_top_29)
 print(corr_top_29_N)
 end if

;*************************************************************************
; Compute Anomaly correlations
;*************************************************************************

 anom_corr = new((/files,3/),float)
 anom_tmax = new((/files,3/),float)
                                    ; Do Loop to capture dates with Tmaxs above a
certain anomaly, that will be used
                                    ; for computing correlations.
 do r=0,15
```

```
  a=0
  do y=0,yrs-1
   do k=0,dys-1
    if (anomaly(3,y,k) .ge. r) then
     delete(anom_corr)
     delete(anom_tmax)
     a=a+1
     anom_corr = new((/files,a/),float)
     anom_tmax = new((/files,a/),float)
    end if
   end do
  end do

  b=0
  do y=0,yrs-1
   do k=0,dys-1
    if(anomaly(3,y,k) .ge. r) then
     do df=0,files-1
      anom_corr(df,b) = anomaly(df,y,k)
      anom_tmax(df,b) = max_array(df,y,k)
     end do
     b=b+1
    end if
   end do
  end do

  corr_anom = new((/files/),float)
  corr_tmax = new((/files/),float)

  a_corr=16
  corr_anom_r = new((/files,a_corr/),float)
  corr_tmax_r = new((/files,a_corr/),float)

  do df=0,files-1                                          ; Do Loop to compute anomaly
and temperature correlations
   corr_anom(df) = escorc(anom_corr(3,:),anom_corr(df,:))
   corr_tmax(df) = escorc(anom_tmax(3,:),anom_tmax(df,:))
   corr_anom_r(df,r) = escorc(anom_corr(3,:),anom_corr(df,:))
   corr_tmax_r(df,r) = escorc(anom_tmax(3,:),anom_tmax(df,:))
  end do

  print("Correlations, Anomaly >= "+r)
  print(corr_anom(:))
  print(corr_tmax(:))
 end do

;*************************************************************************
; Compute Temperature correlations.  All Temps and Temps > a specified temp.
;*************************************************************************

 temp_corr_hi  = new((/files,3/),float)
 temp_corr_all = new((/files,3/),float)
                                     ; Do Loop to capture dates with Tmaxs greater
than a certain temp, that will be used
                                     ; for computing correlations.
  r=100
  s=0
  a=0
  c=0
  do y=0,yrs-1
   do k=0,dys-1
    if (max_array(3,y,k) .ge. r) then
     delete(temp_corr_hi)
     a=a+1
     temp_corr_hi = new((/files,a/),float)
    end if
    if (max_array(3,y,k) .ge. s) then
     delete(temp_corr_all)
     c=c+1
     temp_corr_all = new((/files,c/),float)
    end if
```

```
   end do
  end do
  print(a+" = Number of days above "+r)
  print(c+" = Number of days above "+s)

  b=0
  d=0
  do y=0,yrs-1
   do k=0,dys-1
    if(max_array(3,y,k) .ge. r) then
     do df=0,files-1
      temp_corr_hi(df,b) = max_array(df,y,k)
     end do
     b=b+1
    end if
    if(max_array(3,y,k) .ge. s) then
     do df=0,files-1
      temp_corr_all(df,d) = max_array(df,y,k)
     end do
     d=d+1
    end if
   end do
  end do

 lag=10
 corr_temp_100      = new((/files/),float)
 corr_temp_100_lag  = new((/files,lag+1/),float)
 corr_temp_all      = new((/files/),float)
 corr_temp_all_lag  = new((/files,lag+1/),float)

 do df=0,files-1
   corr_temp_100(df)        = escorc(temp_corr_hi(3,:),temp_corr_hi(df,:))
   corr_temp_100_lag(df,:)  = esccr(temp_corr_hi(3,:),temp_corr_hi(df,:),lag)
   corr_temp_all(df)        = escorc(temp_corr_all(3,:),temp_corr_all(df,:))
   corr_temp_all_lag(df,:)  = esccr(temp_corr_all(3,:), temp_corr_all(df,:),lag)
 end do

  print("Correlations, Sacramento Temps >= "+r)
  print(corr_temp_100(:))
  print("Correlations, Sacramento Temps >= "+s)
  print(corr_temp_all(:))
  print("Lag Correlations, Sacramento Temps >= "+r)
  print(corr_temp_100_lag(:,:))
  print("Lag Correlations, Sacramento Temps >= "+s)
  print(corr_temp_all_lag(:,:))

;*************************************************************
; Create an XY plot of Temperature correlations.
;*************************************************************
 wks13   = gsn_open_wks ("ps","Temp_Corrs_xy")                 ; open workstation

 res13                   = True                      ; plot mods desired
 res13@tiMainString      = "All Max Temperature Correlations"        ; add title
 gsn_define_colormap(wks13,"so4_23")

; res13@xyLineThicknesses = (/1.0,2.0/)              ; make 2nd lines thicker

; res13@xyLineColors      = (/14,13,8,3,2,16,15,11,5,7,1,18,20,22,24,4,6,9,10/)
;
(/"red","orange","green","blue","purple","black","brown","gray","deeppink","aquamarine"/)
;                             ,"","","","","","","","","","")

 res13@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res13@tmXBValues    =
(/0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,2
4.,25. \
                       ,26.,27.,28.,29./)
 res13@tmXBLabels    = (/"Red","RB","Col","Sac","St","Mod","Mer","Fres","Vis","Bkrs" \
                        ,"CC","Eur","Cov","Gr","SF","Mont","SLO","SB","SA","Vist" \

,"Stl","Spok","Yak","Port","Eug","Med","Pend","BkCity","Reno","Tono"/)
```

```
 plot13  = gsn_csm_xy (wks13,ispan(0,29,1),corr_temp_all(:),res13) ; create plot
;*************************************************************************************

 do r=0,a_corr-1
 wks15 = gsn_open_wks ("ps","Anom_Temp_Corrs_xy")
 res15 = True
 gsn_define_colormap(wks15,"so4_23")
 res15@tiMainString = "Temp Correlations on Anomalous Dates"
; res15@xyLineColors = (/14,13,8,3,2,16,15,11,5,7,1,18,20,22,24,4,6,9,10/)
;(/"red","orange","green","blue","purple","black","brown","gray","deeppink","aquamarine"/
)
 res15@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res15@tmXBValues    =
(/0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,2
4.,25. \
                      ,26.,27.,28.,29./)
 res15@tmXBLabels    = (/"Red","RB","Col","Sac","St","Mod","Mer","Fres","Vis","Bkrs" \
                       ,"CC","Eur","Cov","Gr","SF","Mont","SLO","SB","SA","Vist" \

,"Stl","Spok","Yak","Port","Eug","Med","Pend","BkCity","Reno","Tono"/)

 plot15  = gsn_csm_xy (wks15,ispan(0,29,1),corr_tmax_r(:,r),res15) ; create plot
 end do
;************************************************************************
; Create a stacking bar chart of event anomalies, by each year.
; For each year, all city JJAS daily anomalies will be displayed,
; stacked one city on top of the other.
;************************************************************************

 sets = (/"Redding","Red
Bluff","Colusa","Sacramento","Stockton","Modesto","Merced","Fresno","Visalia","Bakersfiel
d" /)
 sets_2 = (/"Crescent City","Eureka","Covelo","Graton","San
Francisco","Monterey","SLO","Santa Barbara","Santa Ana","Vista"/)
 sets_3 = (/"Seattle", "Spokane", "Yakima", "Portland", "Eugene", "Medford", "Pendleton",
"BakerCity", "Reno", "Tonopah"/)
 subset = dys                                    ; Dates, 122 days, June 1 - Aug 31,
y,k=0,121
 plot_years = yrs
 set = dimsizes(sets)
 set_2 = dimsizes(sets_2)
 set_3 = dimsizes(sets_3)
 tick = fspan(0., 121., 122)

 res5          = True
 res5@trYMinF  = -30.0              ; bottom of Y-scale                 ; minimum
anomaly
 res5@trYMaxF  =  30.0              ; nominal top of Y-scale            ; maximum
anomaly
 tickMark      =  10.0               ; tick Mark increment (5 degrees)
 curv_offset   =  res5@trYMaxF-res5@trYMinF   ; range
 res5@trYMaxF  =  curv_offset*set + res5@trYMinF + tickMark

 anom_new   = new((/set,plot_years,subset/),float)
 anom_new_new = new((/set,plot_years,subset/),float)

 do df=0,9                    ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   anom_new(df,y,:) = anomaly(df,y,:) + curv_offset*(9-df)
  end do
 end do

 do df=0,9                      ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   anom_new_new(df,y,:) = anom_new((9-df),y,:)
  end do
 end do

;******
```

```
 res51          = True
 res51@trYMinF  = -30.0             ; bottom of Y-scale                  ; minimum
anomaly
 res51@trYMaxF  =  30.0             ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      =  10.0              ; tick Mark increment (5 degrees)
 curv_offset_2   =  res51@trYMaxF-res51@trYMinF   ; range
 res51@trYMaxF  =  curv_offset_2*set_2 + res51@trYMinF + tickMark

 anom_new_2   = new((/set_2,plot_years,subset/),float)
 anom_new_2_new   = new((/set_2,plot_years,subset/),float)

 do df=10,19                        ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   anom_new_2((df-10),y,:) = anomaly(df,y,:) + curv_offset_2*(19-df)
  end do
 end do

 do df=10,19                        ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   anom_new_2_new((df-10),y,:) = anom_new_2((19-df),y,:)
  end do
 end do

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

 res52          = True
 res52@trYMinF  = -30.0             ; bottom of Y-scale                  ; minimum
anomaly
 res52@trYMaxF  =  30.0             ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      =  10.0              ; tick Mark increment (5 degrees)
 curv_offset_3   =  res52@trYMaxF-res52@trYMinF   ; range
 res52@trYMaxF  =  curv_offset_3*set_3 + res52@trYMinF + tickMark

 anom_new_3   = new((/set_3,plot_years,subset/),float)
 anom_new_3_new   = new((/set_3,plot_years,subset/),float)

 do df=20,29                        ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   anom_new_3((df-20),y,:) = anomaly(df,y,:) + curv_offset_3*(29-df)
  end do
 end do

 do df=20,29                        ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   anom_new_3_new((df-20),y,:) = anom_new_3((29-df),y,:)
  end do
 end do

;*****************************
; Create the actual plot.
;*****************************
 do y=0,yrs-1

 wks5   = gsn_open_wks ("ps", "BarChart_Year"+y)

 res5@xyMonoDashPattern    = True  ; all solid lines
 res5@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res5@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res5@tmYLValues    = fspan(res5@trYMinF,res5@trYMaxF,floattointeger((res5@trYMaxF-
res5@trYMinF)/tickMark)+1 )
 res5@tmYUseLeft    = False
 res5@tmYRLabelsOn  = True

 res5@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res5@tmYRValues    = res5@tmYLValues
 res5@tmYRLabelFontHeightF = res5@tmYLLabelFontHeightF
 res5@tmYLLabels    = (/"-30","-20","-10","0","10","20" \     ; left labels
                      ,"-30","-20","-10","0","10","20" \
                      ,"-30","-20","-10","0","10","20" \
```

```
                      ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20","30",""/)
 res5@tmYRLabels    = (/"","","","Bakersfield","","" \          ; right labels
                       ,"","","","Visalia","","" \
                       ,"","","","Fresno","","" \
                       ,"","","","Merced","","" \
                       ,"","","","Modesto","","" \
                       ,"","","","Stockton","","" \
                       ,"","","","Sacramento","","" \
                       ,"","","","Colusa","","" \
                       ,"","","","Red Bluff","","" \
                       ,"","","","Redding","","","",""/)

 res5@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res5@tiMainString    = "JJAS Daily Anomalies"
 res5@gsnYRefLine = res5@trYMinF + 0.5*curv_offset + ispan(0,set-1,1)*curv_offset

 res5@gsnAboveYRefLineBarColors = "Red"
 res5@gsnBelowYRefLineBarColors = "Blue"
 res5@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res5@tmXBValues    = (/0.,30.,61.,92.,121./)                        ; in IPM,
(/0.,30.,61.,92.,121./)
 res5@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res5@gsnXYBarChart = True
 plot5  = gsn_csm_xy (wks5,tick,anom_new_new(0:9,y,:),res5)
 print("Bar Chart 5 created")

;**********

 wks51   = gsn_open_wks ("ps", "BarChart_2_Year"+y)

 res51@xyMonoDashPattern    = True  ; all solid lines
 res51@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res51@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res51@tmYLValues    = fspan(res51@trYMinF,res51@trYMaxF,floattointeger((res51@trYMaxF-
res51@trYMinF)/tickMark)+1 )
 res51@tmYUseLeft    = False
 res51@tmYRLabelsOn  = True

 res51@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res51@tmYRValues    = res51@tmYLValues
 res51@tmYRLabelFontHeightF = res51@tmYLLabelFontHeightF
 res51@tmYLLabels    = (/"-30","-20","-10","0","10","20" \     ; left labels
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20","30",""/)
 res51@tmYRLabels    = (/"","","","Vista","","" \          ; right labels
                       ,"","","","Santa Ana","","" \
                       ,"","","","Santa Barbara","","" \
                       ,"","","","SLO","","" \
                       ,"","","","Monterey","","" \
                       ,"","","","San Francisco","","" \
                       ,"","","","Graton","","" \
                       ,"","","","Covelo","","" \
                       ,"","","","Eureka","","" \
                       ,"","","","Crescent City","","","",""/)

 res51@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res51@tiMainString    = "JJAS Daily Anomalies"
```

```
 res51@gsnYRefLine = res51@trYMinF + 0.5*curv_offset + ispan(0,set_2-1,1)*curv_offset_2

 res51@gsnAboveYRefLineBarColors = "Red"
 res51@gsnBelowYRefLineBarColors = "Blue"
 res51@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res51@tmXBValues    = (/0.,30.,61.,92.,121./)                        ; in IPM,
(/0.,30.,61.,92.,121./)
 res51@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res51@gsnXYBarChart = True
 plot51  = gsn_csm_xy (wks51,tick,anom_new_2_new(0:9,y,:),res51)
 print("Bar Chart 51 created")

 print("All Bar Charts created")

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

 wks52   = gsn_open_wks ("ps", "BarChart_3_Year"+y)

 res52@xyMonoDashPattern    = True  ; all solid lines
 res52@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res52@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res52@tmYLValues    = fspan(res52@trYMinF,res52@trYMaxF,floattointeger((res52@trYMaxF-
res52@trYMinF)/tickMark)+1 )
 res52@tmYUseLeft    = False
 res52@tmYRLabelsOn  = True

 res52@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res52@tmYRValues    = res52@tmYLValues
 res52@tmYRLabelFontHeightF = res52@tmYLLabelFontHeightF
 res52@tmYLLabels    = (/"-30","-20","-10","0","10","20" \     ; left labels
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20" \
                       ,"-30","-20","-10","0","10","20","30",""/)
 res52@tmYRLabels    = (/"","","","Tonopah","","" \          ; right labels
                       ,"","","","Reno","","" \
                       ,"","","","Baker City","","" \
                       ,"","","","Pendleton","","" \
                       ,"","","","Medford","","" \
                       ,"","","","Eugene","","" \
                       ,"","","","Portland","","" \
                       ,"","","","Yakima","","" \
                       ,"","","","Spokane","","" \
                       ,"","","","Seattle","","","",""/)

 res52@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res52@tiMainString    = "JJAS Daily Anomalies"
 res52@gsnYRefLine = res52@trYMinF + 0.5*curv_offset + ispan(0,set_3-1,1)*curv_offset_3

 res52@gsnAboveYRefLineBarColors = "Red"
 res52@gsnBelowYRefLineBarColors = "Blue"
 res52@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res52@tmXBValues    = (/0.,30.,61.,92.,121./)                        ; in IPM,
(/0.,30.,61.,92.,121./)
 res52@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res52@gsnXYBarChart = True
 plot52  = gsn_csm_xy (wks52,tick,anom_new_3_new(0:9,y,:),res52)
 print("Bar Chart 52 created")

 end do
 print("All Bar Charts created")

;*************************************************************************************
*******
```

```
; Create a stacking bar chart of the number of events each year for each city.
; Cities will be stacked one on top of the other, with year as x-axis and # of events as
y-axis.
;*******************************************************************************
********

 sets = (/"Redding","Red
Bluff","Colusa","Sacramento","Stockton","Modesto","Merced","Fresno","Visalia","Bakersfiel
d"/)
 sets_2 = (/"Crescent City","Eureka","Covelo","Graton","San
Francisco","Monterey","SLO","Santa Barbara","Santa Ana","Vista"/)
 sets_3 = (/"Seattle", "Spokane", "Yakima", "Portland", "Eugene", "Medford", "Pendleton",
"BakerCity", "Reno", "Tonopah"/)
 subset = yrs                                    ; Dates, 122 days, June 1 - Aug 31,
y,k=0,121
 set = dimsizes(sets)
 set_2 = dimsizes(sets_2)
 set_3 = dimsizes(sets_3)
 tick2 = fspan(0., 27., 28)

 res6          = True
 res6@trYMinF  = 0.0             ; bottom of Y-scale                 ; minimum anomaly
 res6@trYMaxF  = 8.0             ; nominal top of Y-scale            ; maximum anomaly
 tickMark      = 2.0              ; tick Mark increment
 curv_offset   =  res6@trYMaxF-res6@trYMinF   ; range
 res6@trYMaxF  =  curv_offset*set + res6@trYMinF + tickMark

 g_total_new = new((/set,subset/),float)
 g_total_new_new = new((/set,subset/),float)

 do df=0,9                       ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   g_total_new(df,y) = g_total(df,y) + curv_offset*(9-df)
  end do
 end do

 do df=0,9                       ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   g_total_new_new(df,y) = g_total_new((9-df),y)
  end do
 end do

;*****

 res61          = True
 res61@trYMinF  = 0.0             ; bottom of Y-scale                 ; minimum anomaly
 res61@trYMaxF  = 8.0             ; nominal top of Y-scale            ; maximum anomaly
 tickMark      = 2.0              ; tick Mark increment
 curv_offset_2   =  res61@trYMaxF-res61@trYMinF   ; range
 res61@trYMaxF  =  curv_offset_2*set_2 + res61@trYMinF + tickMark

 g_total_new_2 = new((/set_2,subset/),float)
 g_total_new_2_new = new((/set_2,subset/),float)

 do df=10,19                      ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   g_total_new_2((df-10),y) = g_total(df,y) + curv_offset_2*(19-df)
  end do
 end do

 do df=10,19                      ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   g_total_new_2_new((df-10),y) = g_total_new_2((19-df),y)
  end do
 end do

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

 res62          = True
 res62@trYMinF  = 0.0             ; bottom of Y-scale                 ; minimum anomaly
 res62@trYMaxF  = 8.0             ; nominal top of Y-scale            ; maximum anomaly
```

```
 tickMark      = 2.0               ; tick Mark increment
 curv_offset_3   =  res62@trYMaxF-res62@trYMinF   ; range
 res62@trYMaxF  =  curv_offset_3*set_3 + res62@trYMinF + tickMark

 g_total_new_3 = new((/set_3,subset/),float)
 g_total_new_3_new = new((/set_3,subset/),float)

 do df=20,29                       ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   g_total_new_3((df-20),y) = g_total(df,y) + curv_offset_3*(29-df)
  end do
 end do

 do df=20,29                       ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   g_total_new_3_new((df-20),y) = g_total_new_3((29-df),y)
  end do
 end do

;*****************************
; Create the actual plot.
;*****************************
 wks6   = gsn_open_wks ("ps", "BarChart_Events")

 res6@xyMonoDashPattern    = True  ; all solid lines
 res6@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res6@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res6@tmYLValues    = fspan(res6@trYMinF,res6@trYMaxF,floattointeger((res6@trYMaxF-
res6@trYMinF)/tickMark)+1 )
 res6@tmYUseLeft    = False
 res6@tmYRLabelsOn  = True

 res6@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res6@tmYRValues    = res6@tmYLValues
 res6@tmYRLabelFontHeightF = res6@tmYLLabelFontHeightF
 res6@tmYLLabels    = (/"0","2","4","6" \     ; left labels
                      ,"0","2","4","6" \
                      ,"0","2","4","6" \
                      ,"0","2","4","6" \
                      ,"0","2","4","6" \
                      ,"0","2","4","6" \
                      ,"0","2","4","6" \
                      ,"0","2","4","6" \
                      ,"0","2","4","6","8",""/)

 res6@tmYRLabels    = (/"","","","Bakersfield","","" \          ; right labels
                      ,"","","","Visalia","","" \
                      ,"","","","Fresno","","" \
                      ,"","","","Merced","","" \
                      ,"","","","Modesto","","" \
                      ,"","","","Stockton","","" \
                      ,"","","","Sacramento","","" \
                      ,"","","","Colusa","","" \
                      ,"","","","Red Bluff","","" \
                      ,"","","","Redding","","","",""/)

 res6@gsnCenterString = "All Cities"
 res6@tiMainString    = "JJAS Yearly Number of Events"
 res6@gsnYRefLine = res6@trYMinF + ispan(0,set-1,1)*curv_offset

; res6@gsnXYBarChartColors = "Red"
 res6@gsnAboveYRefLineBarColors = "Red"
; res6@gsnBelowYRefLineBarColors = "Blue"
 res6@tmXBMode      = "Explicit" ; Define own tick mark labels.
; res6@tmXBValues =
(/0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,2
4.,25.,26.,27./)
; res6@tmXBLabels =
(/"1979","1980","1981","1982","1983","1984","1985","1986","1987","1988","1989","1990","19
```

```
91","1992","1993","1994","1995","1996","1997","1998","1999","2000","2001","2002","2003","
2004","2005","2006"/)

 res6@tmXBLabelFontHeightF = 0.01
 res6@tmXBValues = (/1.,3.,5.,7.,9.,11.,13.,15.,17.,19.,21.,23.,25.,27./)
 res6@tmXBLabels =
(/"1980","1982","1984","1986","1988","1990","1992","1994","1996","1998","2000","2002","20
04","2006"/)


 res6@gsnXYBarChart = True
 plot6  = gsn_csm_xy (wks6,tick2,g_total_new_new(0:9,:),res6)
 print("Plot 6 created")

;*****

 wks61   = gsn_open_wks ("ps", "BarChart_Events_2")

 res61@xyMonoDashPattern    = True  ; all solid lines
 res61@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res61@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res61@tmYLValues    = fspan(res61@trYMinF,res61@trYMaxF,floattointeger((res61@trYMaxF-
res61@trYMinF)/tickMark)+1 )
 res61@tmYUseLeft    = False
 res61@tmYRLabelsOn  = True

 res61@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res61@tmYRValues    = res61@tmYLValues
 res61@tmYRLabelFontHeightF = res61@tmYLLabelFontHeightF
 res61@tmYLLabels    = (/"0","2","4","6" \     ; left labels
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6","8",""/)

 res61@tmYRLabels    = (/"","","","Vista","","" \          ; right labels
                       ,"","","","Santa Ana","","" \
                       ,"","","","Santa Barbara","","" \
                       ,"","","","SLO","","" \
                       ,"","","","Monterey","","" \
                       ,"","","","San Francisco","","" \
                       ,"","","","Graton","","" \
                       ,"","","","Covelo","","" \
                       ,"","","","Eureka","","" \
                       ,"","","","Crescent City","","","","",""/)

 res61@gsnCenterString = "All Cities"
 res61@tiMainString    = "JJAS Yearly Number of Events"
 res61@gsnYRefLine = res61@trYMinF + ispan(0,set_2-1,1)*curv_offset_2

; res6@gsnXYBarChartColors = "Red"
 res61@gsnAboveYRefLineBarColors = "Red"
; res6@gsnBelowYRefLineBarColors = "Blue"
 res61@tmXBMode      = "Explicit" ; Define own tick mark labels.
; res6@tmXBValues =
(/0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,2
4.,25.,26.,27./)
; res6@tmXBLabels =
(/"1979","1980","1981","1982","1983","1984","1985","1986","1987","1988","1989","1990","19
91","1992","1993
;","1994","1995","1996","1997","1998","1999","2000","2001","2002","2003","2004","2005","2
006"/)

 res61@tmXBLabelFontHeightF = 0.01
 res61@tmXBValues = (/1.,3.,5.,7.,9.,11.,13.,15.,17.,19.,21.,23.,25.,27./)
```

```
 res61@tmXBLabels =
(/"1980","1982","1984","1986","1988","1990","1992","1994","1996","1998","2000","2002","20
04","2006"/)

 res61@gsnXYBarChart = True
 plot61  = gsn_csm_xy (wks61,tick2,g_total_new_2_new(0:9,:),res61)

 print("Bar Chart created")
; end do
 print("All Bar Charts created")

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

 wks62   = gsn_open_wks ("ps", "BarChart_Events_3")

 res62@xyMonoDashPattern    = True  ; all solid lines
 res62@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res62@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res62@tmYLValues    = fspan(res62@trYMinF,res62@trYMaxF,floattointeger((res62@trYMaxF-
res62@trYMinF)/tickMark)+1 )
 res62@tmYUseLeft    = False
 res62@tmYRLabelsOn  = True

 res62@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res62@tmYRValues    = res62@tmYLValues
 res62@tmYRLabelFontHeightF = res62@tmYLLabelFontHeightF
 res62@tmYLLabels    = (/"0","2","4","6" \     ; left labels
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6" \
                       ,"0","2","4","6","8",""/)
 res62@tmYRLabels    = (/"Tonopah","","","" \          ; right labels
                       ,"Reno","","","" \
                       ,"Baker City","","","" \
                       ,"Pendleton","","","" \
                       ,"Medford","","","" \
                       ,"Eugene","","","" \
                       ,"Portland","","","" \
                       ,"Yakima","","","" \
                       ,"Spokane","","","" \
                       ,"Seattle","","","","",""/)

 res62@gsnCenterString = "All Cities"
 res62@tiMainString    = "JJAS Yearly Number of Events"
 res62@gsnYRefLine = res62@trYMinF + ispan(0,set_3-1,1)*curv_offset_3

; res6@gsnXYBarChartColors = "Red"
 res62@gsnAboveYRefLineBarColors = "Red"
; res6@gsnBelowYRefLineBarColors = "Blue"
 res62@tmXBMode      = "Explicit" ; Define own tick mark labels.
; res6@tmXBValues =
(/0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,2
4.,25.,26.,27./)
; res6@tmXBLabels =
(/"1979","1980","1981","1982","1983","1984","1985","1986","1987","1988","1989","1990","19
91","1992","1993
;","1994","1995","1996","1997","1998","1999","2000","2001","2002","2003","2004","2005","2
006"/)

 res62@tmXBLabelFontHeightF = 0.01
 res62@tmXBValues = (/1.,3.,5.,7.,9.,11.,13.,15.,17.,19.,21.,23.,25.,27./)
 res62@tmXBLabels =
(/"1980","1982","1984","1986","1988","1990","1992","1994","1996","1998","2000","2002","20
04","2006"/)

 res62@gsnXYBarChart = True
```

```
 plot62  = gsn_csm_xy (wks62,tick2,g_total_new_3_new(0:9,:),res62)

 print("Bar Chart created")
; end do
 print("All Bar Charts created")

;*************************************************************************************
; Create a stacking bar chart of Maximum Temperature of all cities, for each year.
;*************************************************************************************

 sets = (/"Redding","Red
Bluff","Colusa","Sacramento","Stockton","Modesto","Merced","Fresno","Visalia","Bakersfiel
d"/)
 sets_2 = (/"Crescent City","Eureka","Covelo","Graton","San
Francisco","Monterey","SLO","Santa Barbara","Santa Ana","Vista"/)
 sets_3 = (/"Seattle", "Spokane", "Yakima", "Portland", "Eugene", "Medford", "Pendleton",
"BakerCity", "Reno", "Tonopah"/)
 subset = dys                                    ; Dates, 122 days, June 1 - Aug 31,
y,k=0,121
 plot_years = yrs
 set = dimsizes(sets)
 set_2 = dimsizes(sets_2)
 set_3 = dimsizes(sets_3)
 tick = fspan(0., 121., 122)

 print(set)

 res8          = True
 res8@trYMinF  = 75.0              ; bottom of Y-scale                  ; minimum anomaly
 res8@trYMaxF  = 125.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      = 10.0               ; tick Mark increment
 curv_offset   =  res8@trYMaxF-res8@trYMinF   ; range
 res8@trYMaxF  =  curv_offset*set + res8@trYMinF + tickMark

 res81          = True
 res81@trYMinF  = 60.0              ; bottom of Y-scale                  ; minimum
anomaly
 res81@trYMaxF  = 110.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      = 10.0               ; tick Mark increment
 curv_offset_2   =  res81@trYMaxF-res81@trYMinF   ; range
 res81@trYMaxF  =  curv_offset_2*set_2 + res81@trYMinF + tickMark

 res82          = True
 res82@trYMinF  = 60.0              ; bottom of Y-scale                  ; minimum
anomaly
 res82@trYMaxF  = 110.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      = 10.0               ; tick Mark increment
 curv_offset_3   =  res82@trYMaxF-res82@trYMinF   ; range
 res82@trYMaxF  =  curv_offset_3*set_3 + res82@trYMinF + tickMark

; data_chart8 = new((/plot_years,set,subset/),float)
 tmax_new   = new((/set,plot_years,subset/),float)
 tmax_new_new   = new((/set,plot_years,subset/),float)

 do df=0,9                        ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmax_new(df,y,:) = max_array(df,y,:) + curv_offset*(9-df)
  end do
 end do

 do df=0,9                        ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmax_new_new(df,y,:) = tmax_new((9-df),y,:)
  end do
 end do

 tmax_new_2   = new((/set_2,plot_years,subset/),float)
 tmax_new_2_new   = new((/set_2,plot_years,subset/),float)
```

```
 do df=10,19                          ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmax_new_2((df-10),y,:) = max_array(df,y,:) + curv_offset_2*(19-df)
  end do
 end do

 do df=10,19                          ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmax_new_2_new((df-10),y,:) = tmax_new_2((19-df),y,:)
  end do
 end do

 tmax_new_3   = new((/set_3,plot_years,subset/),float)
 tmax_new_3_new   = new((/set_3,plot_years,subset/),float)

 do df=20,29                          ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmax_new_3((df-20),y,:) = max_array(df,y,:) + curv_offset_3*(29-df)
  end do
 end do

 do df=20,29                          ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmax_new_3_new((df-20),y,:) = tmax_new_3((29-df),y,:)
  end do
 end do

;*****************************
; Create the actual plot.
;*****************************
 do y=0,yrs-1

 wks8   = gsn_open_wks ("ps", "BarChart_Tmax_Year"+y)

 res8@xyMonoDashPattern    = True  ; all solid lines
 res8@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res8@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res8@tmYLValues    = fspan(res8@trYMinF,res8@trYMaxF,floattointeger((res8@trYMaxF-
res8@trYMinF)/tickMark)+1 )
 res8@tmYUseLeft    = False
 res8@tmYRLabelsOn  = True

 res8@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res8@tmYRValues    = res8@tmYLValues
 res8@tmYRLabelFontHeightF = res8@tmYLLabelFontHeightF
 res8@tmYLLabels    = (/"75","85","95","105","115" \     ; left labels
                      ,"75","85","95","105","115" \
                      ,"75","85","95","105","115" \
                      ,"75","85","95","105","115" \
                      ,"75","85","95","105","115" \
                      ,"75","85","95","105","115" \
                      ,"75","85","95","105","115" \
                      ,"75","85","95","105","115" \
                      ,"75","85","95","105","115","125",""/)

 res8@tmYRLabels    = (/"Bakersfield","","","","" \          ; right labels
                      ,"Visalia","","","","" \
                      ,"Fresno","","","","" \
                      ,"Merced","","","","" \
                      ,"Modesto","","","","" \
                      ,"Stockton","","","","" \
                      ,"Sacramento","","","","" \
                      ,"Colusa","","","","" \
                      ,"Red Bluff","","","","" \
                      ,"Redding","","","","","",""/)

 res8@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res8@tiMainString    = "JJAS Daily Tmax"
 res8@gsnYRefLine = res8@trYMinF + ispan(0,set-1,1)*curv_offset
```

```
 res8@gsnAboveYRefLineBarColors = "Red"
 res8@gsnBelowYRefLineBarColors = "Blue"
 res8@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res8@tmXBValues    = (/0.,30.,61.,92.,121./)                        ; in IPM,
(/0.,30.,61.,92.,121./)
 res8@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res8@gsnXYBarChart = True
 plot8  = gsn_csm_xy (wks8,tick,tmax_new_new(0:9,y,:),res8)
 print("Bar Chart 8 created")

;*********

 wks81   = gsn_open_wks ("ps", "BarChart_2_Tmax_Year"+y)

 res81@xyMonoDashPattern    = True  ; all solid lines
 res81@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res81@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res81@tmYLValues    = fspan(res81@trYMinF,res81@trYMaxF,floattointeger((res81@trYMaxF-
res81@trYMinF)/tickMark)+1 )
 res81@tmYUseLeft    = False
 res81@tmYRLabelsOn  = True

 res81@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res81@tmYRValues    = res81@tmYLValues
 res81@tmYRLabelFontHeightF = res81@tmYLLabelFontHeightF
 res81@tmYLLabels    = (/"60","70","80","90","100" \     ; left labels
                       ,"60","70","80","90","100" \
                       ,"60","70","80","90","100" \
                       ,"60","70","80","90","100" \
                       ,"60","70","80","90","100" \
                       ,"60","70","80","90","100" \
                       ,"60","70","80","90","100" \
                       ,"60","70","80","90","100" \
                       ,"60","70","80","90","100","110",""/)

 res81@tmYRLabels    = (/"Vista","","","","" \          ; right labels
                       ,"Santa Ana","","","","" \
                       ,"Santa Barbara","","","","" \
                       ,"SLO","","","","" \
                       ,"Monterey","","","","" \
                       ,"San Francisco","","","","" \
                       ,"Graton","","","","" \
                       ,"Covelo","","","","" \
                       ,"Eureka","","","","" \
                       ,"Crescent City","","","","","","",""/)

 res81@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res81@tiMainString    = "JJAS Daily Tmax"
 res81@gsnYRefLine = res81@trYMinF + ispan(0,set_2-1,1)*curv_offset_2

 res81@gsnAboveYRefLineBarColors = "Red"
 res81@gsnBelowYRefLineBarColors = "Blue"
 res81@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res81@tmXBValues    = (/0.,30.,61.,92.,121./)                        ; in IPM,
(/0.,30.,61.,92.,121./)
 res81@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res81@gsnXYBarChart = True
 plot81  = gsn_csm_xy (wks81,tick,tmax_new_2_new(0:9,y,:),res81)
 print("Bar Chart 81 created")

 print("All Bar Charts created")

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

 wks82   = gsn_open_wks ("ps", "BarChart_3_Tmax_Year"+y)

 res82@xyMonoDashPattern    = True  ; all solid lines
```

```
 res82@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res82@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res82@tmYLValues    = fspan(res82@trYMinF,res82@trYMaxF,floattointeger((res82@trYMaxF-
res82@trYMinF)/tickMark)+1 )
 res82@tmYUseLeft    = False
 res82@tmYRLabelsOn  = True

 res82@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res82@tmYRValues    = res82@tmYLValues
 res82@tmYRLabelFontHeightF = res82@tmYLLabelFontHeightF
 res82@tmYLLabels    = (/"60","70","80","90","110" \     ; left labels
                       ,"60","70","80","90","110" \
                       ,"60","70","80","90","110" \
                       ,"60","70","80","90","110" \
                       ,"60","70","80","90","110" \
                       ,"60","70","80","90","110" \
                       ,"60","70","80","90","110" \
                       ,"60","70","80","90","110" \
                       ,"60","70","80","90","100","110",""/)

 res82@tmYRLabels    = (/"Tonopah","","","","" \          ; right labels
                       ,"Reno","","","","" \
                       ,"Baker City","","","","" \
                       ,"Pendleton","","","","" \
                       ,"Medford","","","","" \
                       ,"Eugene","","","","" \
                       ,"Portland","","","","" \
                       ,"Yakima","","","","" \
                       ,"Spokane","","","","" \
                       ,"Seattle","","","","","",""/)

 res82@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res82@tiMainString    = "JJAS Daily Tmax"
 res82@gsnYRefLine = res82@trYMinF + ispan(0,set_2-1,1)*curv_offset_3

 res82@gsnAboveYRefLineBarColors = "Red"
 res82@gsnBelowYRefLineBarColors = "Blue"
 res82@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res82@tmXBValues    = (/0.,30.,61.,92.,121./)                       ; in IPM,
(/0.,30.,61.,92.,121./)
 res82@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res82@gsnXYBarChart = True
 plot82  = gsn_csm_xy (wks82,tick,tmax_new_3_new(0:9,y,:),res82)
 print("Bar Chart 82 created")

 end do
 print("All Bar Charts created")

;*************************************************************************************
*
; Create a stacking bar chart of Minimum Temperatures.
;*************************************************************************************
*

 tmin_min = min(min_array)
 tmin_max = max(min_array)
 print(tmin_min)
 print(tmin_max)

 sets = (/"Redding","Red
Bluff","Colusa","Sacramento","Stockton","Modesto","Merced","Fresno","Visalia","Bakersfiel
d"/)
 sets_2 = (/"Crescent City","Eureka","Covelo","Graton","San
Francisco","Monterey","SLO","Santa Barbara","Santa Ana","Vista"/)
 sets_3 = (/"Seattle", "Spokane", "Yakima", "Portland", "Eugene", "Medford", "Pendleton",
"BakerCity", "Reno", "Tonopah"/)
 subset = dys                                  ; Dates, 122 days, June 1 - Aug 31,
y,k=0,121
 plot_years = yrs
```

```
 set = dimsizes(sets)
 set_2 = dimsizes(sets_2)
 set_3 = dimsizes(sets_3)
 tick = fspan(0., 121., 122)

; print(set)

 res11          = True
 res11@trYMinF  = 50.0              ; bottom of Y-scale                  ; minimum
anomaly
 res11@trYMaxF  = 90.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      = 10.0               ; tick Mark increment
 curv_offset   =  res11@trYMaxF-res11@trYMinF   ; range
 res11@trYMaxF  =  curv_offset*set + res11@trYMinF + tickMark

 res111          = True
 res111@trYMinF  = 50.0              ; bottom of Y-scale                  ; minimum
anomaly
 res111@trYMaxF  = 90.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      = 10.0               ; tick Mark increment
 curv_offset_2   =  res111@trYMaxF-res111@trYMinF   ; range
 res111@trYMaxF  =  curv_offset_2*set_2 + res111@trYMinF + tickMark

 res112          = True
 res112@trYMinF  = 30.0              ; bottom of Y-scale                  ; minimum
anomaly
 res112@trYMaxF  = 70.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark      = 10.0               ; tick Mark increment
 curv_offset_3   =  res112@trYMaxF-res112@trYMinF   ; range
 res112@trYMaxF  =  curv_offset_3*set_3 + res112@trYMinF + tickMark

 tmin_new   = new((/set,plot_years,subset/),float)
 tmin_new_new   = new((/set,plot_years,subset/),float)

 do df=0,9                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmin_new(df,y,:) = min_array(df,y,:) + curv_offset*(9-df)
  end do
 end do

 do df=0,9                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmin_new_new(df,y,:) = tmin_new((9-df),y,:)
  end do
 end do

 tmin_new_2   = new((/set_2,plot_years,subset/),float)
 tmin_new_2_new   = new((/set_2,plot_years,subset/),float)

 do df=10,19                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmin_new_2((df-10),y,:) = min_array(df,y,:) + curv_offset_2*(19-df)
  end do
 end do

 do df=10,19                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   tmin_new_2_new((df-10),y,:) = tmin_new_2((19-df),y,:)
  end do
 end do

 tmin_new_3   = new((/set_3,plot_years,subset/),float)
 tmin_new_3_new   = new((/set_3,plot_years,subset/),float)

 do df=20,files-1                        ; Manipulates data to work with stacking bar
chart
  do y=0,yrs-1
   tmin_new_3((df-20),y,:) = min_array(df,y,:) + curv_offset_3*(29-df)
```

```
  end do
 end do

 do df=20,files-1                        ; Manipulates data to work with stacking bar
chart
  do y=0,yrs-1
   tmin_new_3_new((df-20),y,:) = tmin_new_3((29-df),y,:)
  end do
 end do

;*****************************
; Create the actual plot.
;*****************************
 do y=0,yrs-1

 wks11   = gsn_open_wks ("ps", "BarChart_Tmin_Year"+y)

 res11@xyMonoDashPattern    = True  ; all solid lines
 res11@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res11@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res11@tmYLValues    = fspan(res11@trYMinF,res11@trYMaxF,floattointeger((res11@trYMaxF-
res11@trYMinF)/tickMark)+1 )
 res11@tmYUseLeft    = False
 res11@tmYRLabelsOn  = True

 res11@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res11@tmYRValues    = res11@tmYLValues
 res11@tmYRLabelFontHeightF = res11@tmYLLabelFontHeightF
 res11@tmYLLabels    = (/"50","60","70","80" \     ; left labels
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80","90",""/)

 res11@tmYRLabels    = (/"Bakersfield","","","" \          ; right labels
                       ,"Visalia","","","" \
                       ,"Fresno","","","" \
                       ,"Merced","","","" \
                       ,"Modesto","","","" \
                       ,"Stockton","","","" \
                       ,"Sacramento","","","" \
                       ,"Colusa","","","" \
                       ,"Red Bluff","","","" \
                       ,"Redding","","","","",""/)

 res11@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res11@tiMainString    = "JJAS Daily Minimum Temperature"
 res11@gsnYRefLine = res11@trYMinF + ispan(0,set-1,1)*curv_offset

 res11@gsnAboveYRefLineBarColors = "Red"
 res11@gsnBelowYRefLineBarColors = "Blue"
 res11@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res11@tmXBValues    = (/0.,30.,61.,92.,121./)                        ; in IPM,
(/0.,30.,61.,92.,121./)
 res11@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res11@gsnXYBarChart = True
 plot11  = gsn_csm_xy (wks11,tick,tmin_new_new(0:9,y,:),res11)
 print("Bar Chart 11 created")

;*******

 wks111   = gsn_open_wks ("ps", "BarChart_2_Tmin_Year"+y)

 res111@xyMonoDashPattern    = True  ; all solid lines
 res111@tmYLLabelFontHeightF = 0.01  ; default is 0.02
```

```
 res111@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res111@tmYLValues    =
fspan(res111@trYMinF,res111@trYMaxF,floattointeger((res111@trYMaxF-
res111@trYMinF)/tickMark)+1 )
 res111@tmYUseLeft    = False
 res111@tmYRLabelsOn  = True

 res111@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res111@tmYRValues    = res111@tmYLValues
 res111@tmYRLabelFontHeightF = res111@tmYLLabelFontHeightF
 res111@tmYLLabels    = (/"50","60","70","80" \     ; left labels
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80" \
                       ,"50","60","70","80","90",""/)

 res111@tmYRLabels    = (/"Vista","","","" \          ; right labels
                       ,"Santa Ana","","","" \
                       ,"Santa Barbara","","","" \
                       ,"SLO","","","" \
                       ,"Monterey","","","" \
                       ,"San Francisco","","","" \
                       ,"Graton","","","" \
                       ,"Covelo","","","" \
                       ,"Eureka","","","" \
                       ,"Crescent City","","","","","",""/)

 res111@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res111@tiMainString    = "JJAS Daily Minimum Temperature"
 res111@gsnYRefLine = res111@trYMinF + ispan(0,set_2-1,1)*curv_offset_2

 res111@gsnAboveYRefLineBarColors = "Red"
 res111@gsnBelowYRefLineBarColors = "Blue"
 res111@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res111@tmXBValues    = (/0.,30.,61.,92.,121./)                        ; in IPM,
(/0.,30.,61.,92.,121./)
 res111@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res111@gsnXYBarChart = True
 plot111  = gsn_csm_xy (wks111,tick,tmin_new_2_new(0:9,y,:),res111)
 print("Bar Chart 111 created")

 print("All Bar Charts created")
;*************************************************************

 wks112   = gsn_open_wks ("ps", "BarChart_3_Tmin_Year"+y)

 res112@xyMonoDashPattern    = True  ; all solid lines
 res112@tmYLLabelFontHeightF = 0.01  ; default is 0.02
 res112@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
 res112@tmYLValues    =
fspan(res112@trYMinF,res112@trYMaxF,floattointeger((res112@trYMaxF-
res112@trYMinF)/tickMark)+1 )
 res112@tmYUseLeft    = False
 res112@tmYRLabelsOn  = True

 res112@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
 res112@tmYRValues    = res112@tmYLValues
 res112@tmYRLabelFontHeightF = res112@tmYLLabelFontHeightF
 res112@tmYLLabels    = (/"30","40","50","60" \     ; left labels
                       ,"30","40","50","60" \
                       ,"30","40","50","60" \
                       ,"30","40","50","60" \
                       ,"30","40","50","60" \
                       ,"30","40","50","60" \
```

```
                      ,"30","40","50","60" \
                      ,"30","40","50","60" \
                      ,"30","40","50","60","70",""/)

 res112@tmYRLabels    = (/"Tonopah","","","" \          ; right labels
                      ,"Reno","","","" \
                      ,"Baker City","","","" \
                      ,"Pendleton","","","" \
                      ,"Medford","","","" \
                      ,"Eugene","","","" \
                      ,"Portland","","","" \
                      ,"Yakima","","","" \
                      ,"Spokane","","","" \
                      ,"Seattle","","","","","",""/)

 res112@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
 res112@tiMainString    = "JJAS Daily Minimum Temperature"
 res112@gsnYRefLine = res112@trYMinF + ispan(0,set_3-1,1)*curv_offset_3

 res112@gsnAboveYRefLineBarColors = "Red"
 res112@gsnBelowYRefLineBarColors = "Blue"
 res112@tmXBMode      = "Explicit" ; Define own tick mark labels.
 res112@tmXBValues    = (/0.,30.,61.,92.,121./)                      ; in IPM,
(/0.,30.,61.,92.,121./)
 res112@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

 res112@gsnXYBarChart = True
 plot112  = gsn_csm_xy (wks112,tick,tmin_new_3_new(0:9,y,:),res112)
 print("Bar Chart 112 created")

 end do
 print("All Bar Charts created")


;**********************************************************************************
; Create a stacked bar chart of the DURATION of each event.
;**********************************************************************************

 sets = (/"Redding","Red
Bluff","Colusa","Sacramento","Stockton","Modesto","Merced","Fresno","Visalia","Bakersfiel
d"/)
 sets_2 = (/"Crescent City","Eureka","Covelo","Graton","San
Francisco","Monterey","SLO","Santa Barbara","Santa Ana","Vista"/)
 sets_3 = (/"Seattle", "Spokane", "Yakima", "Portland", "Eugene", "Medford", "Pendleton",
"BakerCity", "Reno", "Tonopah"/)
 subset = dys                                    ; Dates, 122 days, June 1 - Aug 31,
y,k=0,121
 plot_years = yrs
 set = dimsizes(sets)
 set_2 = dimsizes(sets_2)
 set_3 = dimsizes(sets_3)
 tick = fspan(0., 121., 122)

; print(set)

 res10          = True
 res10@trYMinF  = 0.0              ; bottom of Y-scale                  ; minimum anomaly
 res10@trYMaxF  = 10.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark       = 2.0               ; tick Mark increment
 curv_offset    =  res10@trYMaxF-res10@trYMinF   ; range
 res10@trYMaxF  =  curv_offset*set + res10@trYMinF + tickMark

 res101          = True
 res101@trYMinF  = 0.0              ; bottom of Y-scale                  ; minimum
anomaly
 res101@trYMaxF  = 10.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark       = 2.0               ; tick Mark increment
 curv_offset_2    =  res101@trYMaxF-res101@trYMinF   ; range
 res101@trYMaxF  =  curv_offset_2*set_2 + res101@trYMinF + tickMark
```

```
 res102          = True
 res102@trYMinF  = 0.0              ; bottom of Y-scale                  ; minimum
anomaly
 res102@trYMaxF  = 10.0              ; nominal top of Y-scale             ; maximum
anomaly
 tickMark       = 2.0               ; tick Mark increment
 curv_offset_3    =  res102@trYMaxF-res102@trYMinF   ; range
 res102@trYMaxF  =  curv_offset_3*set_3 + res102@trYMinF + tickMark

 num_days_new   = new((/set,plot_years,subset/),float)
 num_days_new_new   = new((/set,plot_years,subset/),float)

 do df=0,9                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   num_days_new(df,y,:) = num_days_events(df,y,:) + curv_offset*(9-df)
  end do
 end do

 do df=0,9                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   num_days_new_new(df,y,:) = num_days_new((9-df),y,:)
  end do
 end do

 num_days_new_2   = new((/set_2,plot_years,subset/),float)
 num_days_new_2_new   = new((/set_2,plot_years,subset/),float)

 do df=10,19                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   num_days_new_2((df-10),y,:) = num_days_events(df,y,:) + curv_offset_2*(19-df)
  end do
 end do

 do df=10,19                         ; Manipulates data to work with stacking bar chart
  do y=0,yrs-1
   num_days_new_2_new((df-10),y,:) = num_days_new_2((19-df),y,:)
  end do
 end do

 num_days_new_3   = new((/set_3,plot_years,subset/),float)
 num_days_new_3_new   = new((/set_3,plot_years,subset/),float)

 do df=20,files-1                        ; Manipulates data to work with stacking bar
chart
  do y=0,yrs-1
   num_days_new_3((df-20),y,:) = num_days_events(df,y,:) + curv_offset_3*(29-df)
  end do
 end do

 do df=20,files-1                        ; Manipulates data to work with stacking bar
chart
  do y=0,yrs-1
   num_days_new_3_new((df-20),y,:) = num_days_new_3((29-df),y,:)
  end do
 end do

;*****************************
; Create the actual plot.
;*****************************
 do y=0,yrs-1

  wks10   = gsn_open_wks ("ps", "BarChart_Duration_"+y)

  res10@xyMonoDashPattern    = True  ; all solid lines
  res10@tmYLLabelFontHeightF = 0.01  ; default is 0.02
  res10@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
  res10@tmYLValues    = fspan(res10@trYMinF,res10@trYMaxF,floattointeger((res10@trYMaxF-
res10@trYMinF)/tickMark)+1 )
  res10@tmYUseLeft    = False
  res10@tmYRLabelsOn  = True
```

```
  res10@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
  res10@tmYRValues    = res10@tmYLValues
  res10@tmYRLabelFontHeightF = res10@tmYLLabelFontHeightF

  res10@tmYLLabels    = (/"0","2","4","6","8" \     ; left labels
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8","10",""/)

  res10@tmYRLabels    = (/"Bakersfield","","","","" \          ; right labels
                       ,"Visalia","","","","" \
                       ,"Fresno","","","","" \
                       ,"Merced","","","","" \
                       ,"Modesto","","","","" \
                       ,"Stockton","","","","" \
                       ,"Sacramento","","","","" \
                       ,"Colusa","","","","" \
                       ,"Red Bluff","","","","" \
                       ,"Redding","","","","","",""/)

  res10@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
  res10@tiMainString    = "Event Duration: Days"
  res10@gsnYRefLine     = res10@trYMinF + ispan(0,set-1,1)*curv_offset

  res10@gsnAboveYRefLineBarColors = "Red"
  res10@gsnBelowYRefLineBarColors = "Blue"
  res10@tmXBMode      = "Explicit"                                ; Define own tick mark
labels.
  res10@tmXBValues    = (/0.,30.,61.,92.,121./)
  res10@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

  res10@gsnXYBarChart = True
  plot10  = gsn_csm_xy (wks10,tick,num_days_new_new(0:9,y,:),res10)
  print("Bar Chart created")

;********

  wks101   = gsn_open_wks ("ps", "BarChart_Duration_2_"+y)

  res101@xyMonoDashPattern    = True  ; all solid lines
  res101@tmYLLabelFontHeightF = 0.01  ; default is 0.02
  res101@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
  res101@tmYLValues    =
fspan(res101@trYMinF,res101@trYMaxF,floattointeger((res101@trYMaxF-
res101@trYMinF)/tickMark)+1 )
  res101@tmYUseLeft    = False
  res101@tmYRLabelsOn  = True

  res101@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
  res101@tmYRValues    = res101@tmYLValues
  res101@tmYRLabelFontHeightF = res101@tmYLLabelFontHeightF

  res101@tmYLLabels    = (/"0","2","4","6","8" \     ; left labels
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8" \
                        ,"0","2","4","6","8","10",""/)

  res101@tmYRLabels    = (/"Vista","","","","" \          ; right labels
```

```
                      ,"Santa Ana","","","","" \
                      ,"Santa Barbara","","","","" \
                      ,"SLO","","","","" \
                      ,"Monterey","","","","" \
                      ,"San Francisco","","","","" \
                      ,"Graton","","","","" \
                      ,"Covelo","","","","" \
                      ,"Eureka","","","","" \
                      ,"Crescent City","","","","","","")

  res101@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
  res101@tiMainString    = "Event Duration: Days"
  res101@gsnYRefLine     = res101@trYMinF + ispan(0,set_2-1,1)*curv_offset_2

  res101@gsnAboveYRefLineBarColors = "Red"
  res101@gsnBelowYRefLineBarColors = "Blue"
  res101@tmXBMode      = "Explicit"                                    ; Define own
tick mark labels.
  res101@tmXBValues    = (/0.,30.,61.,92.,121./)
  res101@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

  res101@gsnXYBarChart = True
  plot101  = gsn_csm_xy (wks101,tick,num_days_new_2_new(0:9,y,:),res101)
  print("Bar Chart created")

 print("All Bar Charts created")

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

  wks102   = gsn_open_wks ("ps", "BarChart_Duration_3_"+y)

  res102@xyMonoDashPattern    = True  ; all solid lines
  res102@tmYLLabelFontHeightF = 0.01  ; default is 0.02
  res102@tmYLMode      = "Explicit"   ; Define own left tick mark labels.
  res102@tmYLValues    =
fspan(res102@trYMinF,res102@trYMaxF,floattointeger((res102@trYMaxF-
res102@trYMinF)/tickMark)+1 )
  res102@tmYUseLeft    = False
  res102@tmYRLabelsOn  = True

  res102@tmYRMode      = "Explicit"   ; Define own right tick mark labels.
  res102@tmYRValues    = res102@tmYLValues
  res102@tmYRLabelFontHeightF = res102@tmYLLabelFontHeightF

  res102@tmYLLabels    = (/"0","2","4","6","8" \     ; left labels
                       ,"0","2","4","6","8" \
                       ,"0","2","4","6","8" \
                       ,"0","2","4","6","8" \
                       ,"0","2","4","6","8" \
                       ,"0","2","4","6","8" \
                       ,"0","2","4","6","8" \
                       ,"0","2","4","6","8" \
                       ,"0","2","4","6","8","10",""/)

  res102@tmYRLabels    = (/"Tonopah","","","","" \          ; right labels
                      ,"Reno","","","","" \
                      ,"Baker City","","","","" \
                      ,"Pendleton","","","","" \
                      ,"Medford","","","","" \
                      ,"Eugene","","","","" \
                      ,"Portland","","","","" \
                      ,"Yakima","","","","" \
                      ,"Spokane","","","","" \
                      ,"Seattle","","","","","","")

  res102@gsnCenterString = "Year " + sprinti("%0.4i", y+1979)
  res102@tiMainString    = "Event Duration: Days"
  res102@gsnYRefLine     = res102@trYMinF + ispan(0,set_3-1,1)*curv_offset_3

  res102@gsnAboveYRefLineBarColors = "Red"
```

```
  res102@gsnBelowYRefLineBarColors = "Blue"
  res102@tmXBMode      = "Explicit"                                    ; Define own
tick mark labels.
  res102@tmXBValues    = (/0.,30.,61.,92.,121./)
  res102@tmXBLabels    = (/"1June","1July","1Aug","1Sept","30Sept"/)

  res102@gsnXYBarChart = True
  plot102  = gsn_csm_xy (wks102,tick,num_days_new_3_new(0:9,y,:),res102)
  print("Bar Chart created")

 end do
 print("All Bar Charts created")

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

end
```

IPM_Stats.ncl

```
;*************************************************************************
; This program determines matches of heat wave event dates between Sacramento
; and the other stations, based on the different ranking methods
; for the HW events in the cities specified in 'datafiles'.
;*************************************************************************
; CHANGE: Reading in of Top_Event_Dates, Top_Event_Dates_N, or Top_Event_Dates_max
; to correspond with the Top Events ranking based on the highest average maximum
; temperature, the highest 3-day average of maximum temperature anomalies, or the
; highest event maximum temperature.
;*************************************************************************
'

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/shea_util.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

;---------------------------------------------------------------

begin

 top_rank=15          ; corresponds to the top 15 ranked heat wave events

;*******************************************
; Read in all Event dates arrays for each city.
;*******************************************
 data1=asciiread("Redding_event_dates_array", -1, "integer")
 Redding_date = data1(:)
 print(Redding_date)

 data2=asciiread("RedBluff_event_dates_array", -1, "integer")
 RedBluff_date = data2(:)
 print(RedBluff_date)

*
*
*** This is performed this way for every city *****
*
*

 data30=asciiread("Tonopah_event_dates_array", -1, "integer")
 Tonopah_date = data30(:)
 print(Tonopah_date)

 print("data files read")
;*************************************************************************
; Read in Top Event Dates based on the HW definition requiring the average
; temperature of the event to be >= 100 degrees
;*************************************************************************

; top1=asciiread("Redding_Top_Event_Dates", -1, "integer")
; Redding_top = top1(:)
; print(Redding_top)

; top2=asciiread("RedBluff_Top_Event_Dates", -1, "integer")
; RedBluff_top = top2(:)
; print(RedBluff_top)

*
*
*** (This is performed this way for every city) *****
*
*

; top30=asciiread("Tonopah_Top_Event_Dates", -1, "integer")
; Tonopah_top = top30(:)
; print(Tonopah_top)

;*************************************************************************
'
```

```
; Read in Top Event Dates based on the highest 3-day anomaly average ranking.
;*************************************************************************
;
; top1=asciiread("Redding_Top_Event_Dates_N", -1, "integer")
; Redding_top = top1(:)
; print(Redding_top)

; top2=asciiread("RedBluff_Top_Event_Dates_N", -1, "integer")
; RedBluff_top = top2(:)
; print(RedBluff_top)

*
*
*** (This is performed this way for every city) ***
*
*

; top30=asciiread("Tonopah_Top_Event_Dates_N", -1, "integer")
; Tonopah_top = top30(:)
; print(Tonopah_top)

;*************************************************************
; Read in Top Event Dates based on the Maximum Temperature ranking.
;*************************************************************
;
 top1=asciiread("Redding_Top_Event_Dates_max", -1, "integer")
 Redding_top = top1(:)
 print(Redding_top)

 top2=asciiread("RedBluff_Top_Event_Dates_max", -1, "integer")
 RedBluff_top = top2(:)
 print(RedBluff_top)

*
*
*** (This is performed this way for every city) ***
*
*

 top30=asciiread("Tonopah_Top_Event_Dates_max", -1, "integer")
 Tonopah_top = top30(:)
 print(Tonopah_top)

 print("top files read")

;*******************************************************
; Determine HW Event Matches of all heat wave event dates.
;*******************************************************
;
; Initialize date array with a random number.
; If the date of Sacramento HW mathces the date of
; another city's HW, then declare that a match.
; Reinitialize array with the number of matching dates.
; Repeat with each city.
;*******************************************************


 Sac_Redding_date = new((/10/),integer)             ; initialize array with random number

 m=0
 do i=0,dimsizes(Sac_date)-1
  do j=0,dimsizes(Redding_date)-1
   if (Sac_date(i) .eq. Redding_date(j)) then       ; if the dates of a Sac HW mathches
the date of
    delete(Sac_Redding_date)
    m=m+1
    Sac_Redding_date = new((/m/),integer)
   end if
  end do
 end do

 m=0
```

```
 do i=0,dimsizes(Sac_date)-1
  do j=0,dimsizes(Redding_date)-1
   if (Sac_date(i) .eq. Redding_date(j)) then
    Sac_Redding_date(m) = Sac_date(i)
    m=m+1
   end if
  end do
 end do
 match = m
 print(Sac_Redding_date(:))
 print("Redding: "+m+" matches")

*
*
**** (This is performed this way for every station)***
*
*

 Sac_Tonopah_date = new((/10/),integer)

 m=0
 do i=0,dimsizes(Sac_date)-1
  do j=0,dimsizes(Tonopah_date)-1
   if (Sac_date(i) .eq. Tonopah_date(j)) then
    delete(Sac_Tonopah_date)
    m=m+1
    Sac_Tonopah_date = new((/m/),integer)
   end if
  end do
 end do

 m=0
 do i=0,dimsizes(Sac_date)-1
  do j=0,dimsizes(Tonopah_date)-1
   if (Sac_date(i) .eq. Tonopah_date(j)) then
    Sac_Tonopah_date(m) = Sac_date(i)
    m=m+1
   end if
  end do
 end do
 match = m
 print(Sac_Tonopah_date(:))
 print("Tonopah: "+m+" matches")

;************************************************************************
; Determine HW Event Matches of the Top N Events
;************************************************************************

 print("Top "+top_rank+" Events Matches")

 Sac_Redding_top = new((/10/),integer)

 m=0
 do i=0,dimsizes(Sac_top)-1
  if (Sac_top(i) .ne. 0) then
   do j=0,dimsizes(Redding_top)-1
    if (Redding_top(j) .ne. 0) then
     if (Sac_top(i) .eq. Redding_top(j)) then
      delete(Sac_Redding_top)
      m=m+1
      Sac_Redding_top = new((/m/),integer)
     end if
    end if
   end do
  end if
 end do

 m=0
 do i=0,dimsizes(Sac_top)-1
  if (Sac_top(i) .ne. 0) then
   do j=0,dimsizes(Redding_top)-1
```

```
    if (Redding_top(j) .ne. 0) then
     if (Sac_top(i) .eq. Redding_top(j)) then
      Sac_Redding_top(m) = Sac_top(i)
      m=m+1
     end if
    end if
   end do
  end if
 end do
 match = m
 print(Sac_Redding_top(:))
 print("Redding: "+m+" matches")

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

*
*** (Performed this way for every state) ***
*

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

 Sac_Tonopah_top = new((/10/),integer)

 m=0
 do i=0,dimsizes(Sac_top)-1
  if (Sac_top(i) .ne. 0) then
  do j=0,dimsizes(Tonopah_top)-1
  if (Tonopah_top(j) .ne. 0) then
   if (Sac_top(i) .eq. Tonopah_top(j)) then
    delete(Sac_Tonopah_top)
    m=m+1
    Sac_Tonopah_top = new((/m/),integer)
   end if
 end if
  end do
 end if
 end do

 m=0
 do i=0,dimsizes(Sac_top)-1
  if (Sac_top(i) .ne. 0) then
  do j=0,dimsizes(Tonopah_top)-1
  if (Tonopah_top(j) .ne. 0) then
   if (Sac_top(i) .eq. Tonopah_top(j)) then
    Sac_Tonopah_top(m) = Sac_top(i)
    m=m+1
   end if
 end if
  end do
 end if
 end do
 match = m
 print(Sac_Tonopah_top(:))
 print("Tonopah: "+m+" matches")

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


end
```

rngen.ncl

```
;**********************************************************************************************
*****************************
; This program creates 1000 random ensembles, with 15 random members in each ensemble,
; for each station, using rand().
; Each random ensemble member corresponds to a maximum JJAS temperature.
; The averages of the random ensembles are compared to the one 'target' ensemble
; of maximum temperature on the onset dates of Sacreamento's 15 hottest heat wave events.
;**********************************************************************************************
*****************************
; CHANGE:
; 1) Asciiread statements to read in either top heat wave events ranked by
;    1) highest consecutive 3-day average of max temp anomalies, or by
;    2) highest event maximum temperature.
;
; 2) Histogram Main Title and Center Title to correspond to the data that is read in.
;**********************************************************************************************
*****************************

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/shea_util.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

;---------------------------------------------------------------

begin

 files     = 30                      ; 30 files corresponding to the 30 stations
 yrs       = 28                      ; 28 years in dataset
 dys       = 122              ; 122 day in JJAS
 JJAS_days = 3416             ; 3416 days in 28 years of JJAS
 num_col   = 3                ; column 1 = time, 2 = max temps, 3 = min temps
 num_time  = 10135            ; 10135 total times in 28 full years

 ensembles  = 1000            ; number of random ensembles to be created
 num_in_ens = 15              ; number of random members in random ensembles

 low  = 0                     ; random numbers to be created between 0 and 1
 high = 1
 con = (high - low)/32766     ; note: random number generator [rand()] creates random
number between 0 and 32766

 random = new((/files,ensembles,num_in_ens/),integer)   ; random number array
 Rn = new((/files,ensembles,num_in_ens/),float)             ; array of random numbers
 Rn_2 = new((/files,ensembles,num_in_ens/),float)    ; random number index array
 Rn_Index = new((/files,ensembles,num_in_ens/),float)       ; array of random number
indexes, from 0 to 3416
 Rn_Tmax = new((/files,ensembles,num_in_ens/),float) ; array of random Tmaxs corresponding
to 0,3416 index
 ens_ave = new((/files,ensembles+1/),float)         ; array of ensemble averages
 ens_ave_min = new((/files/),float)                 ; min value of the 1,000 random
ensembles for each station
 ens_ave_max = new((/files/),float)                 ; max value of the 1,000 random
ensembles for each station
 ens_ave_min_int = new((/files/),integer)           ; array of ensemble mins converted to
integers
 ens_ave_max_int = new((/files/),integer)           ; array of ensemble maxs convereted
to integeres
 ens_ave_perm = new((/files,ensembles+1/),integer)  ; array of ensemble average
permutations
 ens_ave_perm_5 = new((/files/),float)                      ; array of 5th top randomly
created value of each station
 ens_ave_perm_6 = new((/files/),float)                      ; array of 6th top randomly
created value of each station
 ens_ave_995    = new((/files/),float)                      ; array of the 99.5% value
for each station
 target_place   = new((/files/),string)                     ; array of the values at
which the target ensemble means lie
```

```
 datafiles = (/"Redding", "RedBluff", "Colusa", "Sac", "Stockton", "Modesto", "Merced",
"Fresno", "Visalia", "Bkrsfld" \
             , "CrescentCity", "Eureka", "Covelo", "Graton", "SanFrancisco", "Monterey",
"SLO", "SantaBarbara", "SantaAna" \
            , "Vista" , "Seattle", "Spokane", "Yakima", "Portland", "Eugene", "Medford",
"Pendleton", "BakerCity", "Reno" \
           , "Tonopah"/)

;*******************************************************************************
; Read in target ensemble averages of filtered/unfiltered data(fil/UNfil),
; with/without the average max temp requirement in the HW definition (tave/notave),
; and according to the average max temp ranking(ave), the max temp ranking(max),
; or the highest consecutive 3-day anomaly average ranking(Nday).
;*******************************************************************************

; data_target_ave = asciiread("target_ave_UNfil_notave",-1,"float") ; Target Ave based on
average maximum temperature,
; data_target_ave = asciiread("target_ave_Nday_UNfil_notave",-1,"float"); Target Ave
based on 3-day anomaly average ranking
; data_target_ave = asciiread("target_ave_max_UNfil_notave",-1,"float")  ; Target Average
based on max temperature ranking

; data_target_ave = asciiread("target_ave_UNfil_tave",-1,"float")     ; Target Ave based
on average maximum temperature
; data_target_ave = asciiread("target_ave_Nday_UNfil_tave",-1,"float"); Target Ave based
on 3-day anomaly average ranking
; data_target_ave = asciiread("target_ave_max_UNfil_tave",-1,"float")  ; Target Average
based on max temperature ranking

; data_target_ave = asciiread("target_ave_fil_notave",-1,"float")     ; Target Ave based
on average maximum temperature

 data_target_ave = asciiread("Sacdate_target_3day_avg_unfil",-1,"float") ; Target Ave
based on 3-day anomaly average ranking
; data_target_ave = asciiread("Sacdate_target_max_avg_unfil",-1,"float")  ; Target
Average based on max temperature ranking

; data_target_ave = asciiread("Sacdate_target_3day_avg_fil",-1,"float")  ; Target average
based on 3-day anomaly avg ranking
; data_target_ave = asciiread("Sacdate_target_max_avg_fil",-1,"float")

 do df=0,files-1
  print(df)
  data_max          = asciiread(datafiles(df)+"_JJAS_max",-1,"float")             ; read
in UNfiltered temp data
;  data_max          = asciiread(datafiles(df)+"_JJAS_max_fil",-1,"float")      ; read in
FILTERED temp data

  srand(123456789)            ; sets the seed for the random number generator rand()

  do i=0,ensembles-1                                ; cycles through all ensembles
   do j=0,num_in_ens-1                              ; cycles through all members
    random(df,i,j) = rand()                         ; rand() is random number generator
    Rn(df,i,j) = random(df,i,j) * 3415                     ; random number created
    Rn_2(df,i,j) = Rn(df,i,j)/32766                 ; 0,3416 index created
    Rn_Index(df,i,j) = round(Rn_2(df,i,j),3)               ; round index to the nearest
whole number
    Rn_Tmax(df,i,j) = data_max(floattointeger(Rn_Index(df,i,j)))   ; Tmax for particular
index retrieved
   end do
   ens_ave(df,i) = avg(Rn_Tmax(df,i,:))                    ; ensemble average for the
particular ensemble is calculated
  end do
  ens_ave(df,ensembles) = data_target_ave(df)       ; sets the target ensemble average
for each station as the last entry
  print(ens_ave(df,ensembles))
  print("df="+df+" done")

  ens_ave_min(df) = min(ens_ave(df,:))              ; calculates the minimum of the
ensemble averages array
```

```
  ens_ave_max(df) = max(ens_ave(df,:))              ; calculates the maximum of the
ensemble averages array
  ens_ave_min_int(df) = floattointeger(ens_ave_min(df))
  ens_ave_max_int(df) = round(ens_ave_max(df),3) + 1
  ens_ave_perm(df,:) = dim_pqsort(ens_ave(df,:),-1)         ; arrange ensemble averages
in descending order
  ens_ave_perm_5(df)     = ens_ave(df,ens_ave_perm(df,4))   ; grab the 5th ensemble
average value
  ens_ave_perm_6(df)     = ens_ave(df,ens_ave_perm(df,5))   ; grab the 6th ensemble
average value
  ens_ave_995(df)        = (( (ens_ave_perm_5(df)) + (ens_ave_perm_6(df)) ) / 2)   ; find
the value b/w the 5th and 6th values

                                                    ; Label target ensemble avgerage as
satisfying or not satisfying the 99.5% stat
                                                    ; based on if it falls above or below
the 99.5% value
  if (data_target_ave(df) .ge. ens_ave_995(df)) then
   target_place(df) = "SATISFIES"
  else
   target_place(df) = "DOES NOT SATISFY"
  end if

 end do

 print(data_target_ave)
 print(ens_ave)
 print(ens_ave_min)
 print(ens_ave_max)
 print(ens_ave_min_int)
 print(ens_ave_max_int)
 print(ens_ave_perm)
 print(ens_ave_perm_5)
 print(ens_ave_perm_6)
 print(ens_ave_995)
 print("All done")

 do df=0,files-1
  print(datafiles(df)+": 99.5% of values less than "+ens_ave_995(df)+" degrees")
  print(datafiles(df)+": Target Ensemble Average = "+data_target_ave(df)+" degrees, and
"+target_place(df)+" the 99.5% test")
 end do

;*************************************************************************
; Create histogram of random ensembles
;*************************************************************************

 do df=0,files-1
  wks = gsn_open_wks("ps","histogram_"+datafiles(df))               ; open workstation
  gsn_define_colormap(wks,"rainbow")               ; choose colormap

  res                           = True          ; plot mods desired
 res@tiMainString = datafiles(df)
;  res@tiMainString = datafiles(df)+" JJAS Max Temp Ensemble Averages :C: Max Temp
Ranking Scheme, Unfiltered, No Tave"
;   res@tiMainString = datafiles(df)+" JJAS Max Temp Ensemble Averages :C: 3-Day Ranking
Scheme, Unfiltered, No Tave"
;   res@tiMainString = datafiles(df)+" JJAS Max Temp Ensemble Averages :C: 3-Day Ranking
Scheme, Filtered, No Tave"
;  res@tiCenterString = "3-day Ranking Scheme"
;  res@tiCenterString = "Average Max Temp Ranking Scheme"
;  res@tiCenterString = "Max Temp Ranking Scheme, Unfiltered, No Tave"
;  res@tiHistogramSelectNiceIntervals = True
  res@gsnHistogramBinWidth = 0.5
  res@tmXBLabelStride = 2
  res@tmXBLabelFontHeightF = .01

  res@gsnDraw       = False          ; do not draw
  res@gsnFrame      = False          ; do not advance

  plot=gsn_histogram(wks,ens_ave(df,:),res)                 ; create histogram
```

```
  txres               = True
  txres@txFontHeightF = 0.05             ; Set the font height

;  label = "********"

;
text=gsn_add_text(wks,plot,label,tmXBValues(floattoint(ens_ave(df,ensembles))),0.2*(float
toint(tmYLValues(nTmY-1))),txres)
 gsn_text(wks,plot,"*",floattointeger(data_target_ave(df)),5.0,txres)

 draw(plot)
 frame(wks)

 end do

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

end
```

NCL_mapping.ncl

```
;**********************************************************************************
; This program spatially plots station data onto a CA, OR, WA, and NV map,
; according to the latitude/longitude coordinates of each station.
;**********************************************************************************
; CHANGE: 1) Title of Map (res@tiMainString), corresponding to data plotted
;         2) Data to be plotted (gsn_text)
;**********************************************************************************

load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/shea_util.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

;---------------------------------------------------------------

begin

;************************************************
; based on the ncl demo program ce_3.ncl
;************************************************
;
;************************************************
;
; create plot
;************************************************
  wks = gsn_open_wks("ps" ,"NCL_CA_map")          ; open a ps file

 res                       = True     ; plot mods desired
 res@gsnFrame              = False       ; Do not advance frome
 res@mpFillOn              =  False          ; turn off map fill
 res@mpOutlineDrawOrder    = "PostDraw"      ; draw continental outline last
 res@mpOutlineBoundarySets = "GeophysicalAndUSStates" ; state boundaries

 res@pmTickMarkDisplayMode = "Always"; use NCL default lat/lon labels
 res@tiMainFontHeightF = 0.02

; res@tiMainString         = "California, Washington, Oregon, and Nevada Stations"    ;
plot title
; res@tiMainString = " CA, WA, OR, and NV HW Event:C:Averaged Normalized Anomalies"
 res@tiMainString = " CA, WA, OR, and NV HW Event Averaged Normalized:C:Anomalies on Top
Sac 15 3-Day Anom Avg Dates, Unfil"
; res@tiMainString = "         CA, WA, OR, and NV:C: JJAS Average Maximum Temps"
; res@tiMainString = "       HW Event Matches..Unfiltered,:C:No Avg Max Temp, 3-Day Anom
Avg Rank"
; res@tiMainString="  HW Event Matches..Unfiltered,:C:Avg Max Temp, Max Temp Rank"
; res@tiMainString="     HW Event Matches..Unfiltered,:C:No Avg Max Temp, Max Temp Rank"
; res@tiMainString="     HW Event Matches..Unfiltered,:C:Avg Max Temp, 3-Day Anom Avg
Rank"
; res@tiMainString        = "CA, WA, OR, and NV JJAS Max Temp Standard Deviations"
; res@tiMainString        = "CA, WA, OR, and NV Average Minimum Temperatures"
; res@tiMainString        = "CA, WA, OR, and NV Average Maximum Temperatures"

; note that the gsn_csm_*map_ce templates automatically set
; res@mpLimitMode="LatLon" for you. If you are plotting a different
; projection, you may have to set this resource.

 res@mpMinLatF            = 32       ; Southernmost latitude
 res@mpMaxLatF            = 49.             ; Northernmost latitude
 res@mpMinLonF            = -125.    ; Westernmost longitude
 res@mpMaxLonF            = -116.    ; Easternmost longitude

 plot  = gsn_csm_map_ce(wks,res)

 draw (plot)

;***************Text Resource*****************************************

 txres               = True                     ; text mods desired
 txres@txFontHeightF = 0.011                     ; font smaller. default big
 txres@txFontThicknessF = 2.0
```

```
 txres@txFontColor = "black"
 txres@txJust        = "CenterCenter"

; add some text at specific locations can use integer or floating latitudes
;   enter lon,lat of station.
; gsn_text(wks,plot,"+",-121.3,48,txres)
; gsn_text(wks,plot,"*",-121.3,38.5,txres)
; tests of encoding a number:
; sum1 = 101.
; gsn_text (wks,plot,"T="+sum1, -122.0, 41. ,txres)
; sum2 = 105.
; gsn_text (wks,plot,sum2, -120.0, 35. ,txres)
; sum3 = 0.923
; gsn_text (wks,plot,sum3, -118.0, 47. ,txres)

 files=30

 cities = (/"RDD","RBL","COL","SMF","SCK","MOD","MCE","FAT","VIS","BFL" \  ; 3-letter
Station codes
           ,"CEC","EKA","COV","GRT","SFO","MRY","SBP","SBA","SNA","VST" \
           ,"SEA","GEG","YKM","PDX","EUG","MFR","PDT","BKE","RNO","TPH" /)

 norm_anom_avg = (/"1.63","1.65","1.94","1.75","1.95","1.87","1.97","1.75","1.92","1.80"
\       ; Station HW event avg normalized anoms
                 ,"3.28","4.23","1.67","1.97","2.56","2.63","2.27","3.41","2.52","2.37"
\

,"2.02","1.50","1.62","1.83","1.71","1.60","1.52","1.45","1.66","1.69"/)

 Sacdates_norm_anom_avg =
(/"1.47","1.58","1.62","1.89","1.79","1.59","1.25","1.26","1.14","1.19" \  ; avg
normalized anomalies on Sac HW event dates

,"0.76","0.19","1.26","1.50","1.66","1.69","1.29","0.29","0.93","0.94" \

,"0.96","0.57","0.74","0.94","0.88","1.09","0.75","0.54","0.81","0.77"/)

 tmax_avg = (/"94.1","93.3","91.0","89.7","90.7","90.8","93.7","94.7","91.3","94.2" \
       ; average JJAS Tmax
             ,"65.3","63.8","89.6","83.0","72.3","68.9","79.1","74.7","82.3","81.7" \
             ,"74.5","80.3","84.5","79.9","80.7","88.4","84.2","82.1","88.0","87.5"/)

 matches_unfil_notave_3day = (/"5","6","6","*","9","6","3","3","3","5" \   ; # of matches
of HW event based on unfiltered, no avg temp requirement,
                              ,"0","0","5","6","5","4","3","1","4","3" \   ; 3day avg
anom ranking HW events
                              ,"2","2","3","1","3","2","1","2","1","1"/)

 matches_unfil_notave_max = (/"6","8","4","*","6","5","1","1","1","2" \           ; # of
matches of HW events based on unfiltered, no avg temp requirement,
                             ,"0","0","4","4","3","1","2","2","2","1" \           ; max
temp ranking HW events
                             ,"3","1","3","3","2","2","2","1","1","0"/)

 matches_unfil_tave_3day = (/"5","7","5","*","7","5","3","2","2","4" \            ; # of
matches of HW events based on unfiltered, avg temp requirement,
                              ,"0","0","4","4","0","0","0","0","1","1" \           ; 3day
avg anom ranking HW events
                             ,"0","0","1","1","2","4","2","0","0","0"/)

 matches_unfil_tave_max = (/"7","8","3","*","6","4","2","0","0","1" \             ; # of
matches of HW events based on unfiltered, avg temp requirement,
                            ,"0","0","4","1","0","0","0","0","2","1" \            ; max
temp ranking HW events
                            ,"0","0","2","1","1","2","2","0","0","0"/)

 latitude  = (/40.52, 40.15, 39.2, 38.52, 38., 37.65, 37.28, 36.77, 36.33, 35.42 \;
latitude coordinates for each station
             , 41.77, 40.9, 39.78, 38.43, 37.62, 36.6, 35.3, 34.42, 33.75, 33.23 \
             , 47.45, 47.62, 46.57, 45.4, 44.13, 42.38, 45.70, 44.85, 39.48, 38.07 /)
```

```
 longitude = (/-122.32, -122.25, -122.02, -121.5, -121.32, -121., -120.52, -119.72, -
119.3, -119.05 \       ; longitude coordinates for each station
             , -124.5, -124.5, -123.25, -122.5, -123.0, -122.3, -121.25, -119.0, -117.25,
-117.0 \
             , -121.9, -117.53, -120.55, -122.60, -123.22, -122.87, -118.85, -117.82, -
119.5, -117.02 /)

 do i=0,files-1
;  gsn_text (wks, plot, cities(i), longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, norm_anom_avg(i), longitude(i), latitude(i), txres)
  gsn_text (wks, plot, Sacdates_norm_anom_avg(i), longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, tmax_avg(i), longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, matches_unfil_notave_3day(i), longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, matches_unfil_tave_max(i), longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, matches_unfil_notave_max(i), longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, matches_unfil_tave_3day(i), longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, " "+JJAS_SD(i)+" ", longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, " "+tmax_avg(i)+" ", longitude(i), latitude(i), txres)
;  gsn_text (wks, plot, " "+tmin_avg(i)+" ", longitude(i), latitude(i), txres)
 end do

 frame(wks)
;*********************************************


end
```