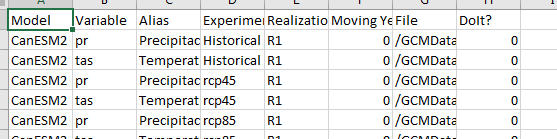
*Basic Instructions for Simple, Single-Region Basin:*

**Defining Variable and Function Arguments:**

* Ensure that all packages in the function library and batch processing scripts are installed
* Set the path in the batch processing script to the location of the function library
* Set the working directory to the location of the “Archive\_list\_batch…csv” file
* Define the main folder (mFolder) path, from which you will define other locations, such as locations of historical record files, results folder path etc.
* Define paths of the Historical Observation record files (daily, monthly, and the stations Catalog, which has data such as Lat, Lan, Altitude etc..). These files should be named in the following manner: “SERIES\_pr\_DIARIO\_v4.csv”, where “pr” value (could be anything, like “tas” as well) corresponds to the variable names in the “Variable” column of the excel “Archive\_list\_batch…csv” table:
* 
* Define path of the background GIS shapefile of your basin to be used in the output graphics
* Define the results folder (rFolder) path, where you’d like to script to output all the graphs, Rdata objects, and .csv time-series files produced
* Define the bounding box (bbox) in decimal degrees (-180 to 180) of your study region

*Funciton Arguments (only need to modify those specified below)*

For many functions, there are arguments for “minObsYear”, “maxObsYear”, “minGCMYear”, “maxGCMYear”. These should all be the same between functions and represent the first and last years of the observed record you are analyzing (minObsYear & maxObsYear), and the first and last years of the FUTURE GCM time-series you’d like to analyze (minGCMYear & maxGCMYear)

**compare\_GCM**

* + *alignHistYears –* **TRUE** will only analyze historic observed and historic GCM records for the same number of years (RECOMMENDED). **FALSE** will use all available data of each data set, but they will likely not represent common time periods.

**knn\_bootstrap**

* + *bootstrapLength –* the number of future years of the GCM models being used to create synthetic series (i.e. 2015 – 2040 = 26 years)
  + *nearWindow –* number of days before and after each given day in the future scenario to calculate transition matrices (recommended to leave at 15)
  + *JPmode –* Defines whether transition matrices will be calculated according to a yearly, monthly, or moving window scheme, recommended toleave on “Window”
  + alignHistYears – recommended to leave on **TRUE**
  + HistRepro – **TRUE** indicated that the synthetic series will be a historical reproduction, or that the transition matrices being used to synthesize the bootstrapped time-series are merely those calculated from the observed historic time-series, and not using the GCM models at all. This is to assess how well the tool is performing to reproduce historical series. **FALSE** is what you should use when generating actual climate change bootstrapped series.

**sinteticSeries**

* + *fullObsCatalog* – unless you are dealing with a very large basin and are hoping to stitch together observed records, each with a unique extremes-correction performed, then it is recommended to just leave this as **FALSE**
  + *HistRepro –* should agree with what is defined for this argument in the knn\_bootstrap function

**CHANGES/IMPROVEMENTS TO TOOL**

Code Change Log (July 18 2017 – N. Depsky)

***GCMClimTool\_function\_library:***

* Various New packages required
* Added Value to Percentile function – *val2pctl*
* Added Linear Interpolation function – *linapprox* (for use in extremes correction)
* Added function to find Mode of a dataset – *getmode* (for improved climate metrics output)
* Additional metrics output by *wet\_dry\_spell\_length* function
* Added *tri\_state\_joint\_prob\_monthly* function

***read\_GCM\_NCDF:***

* added **sta\_bbox** = “region” | “pixel” argument
* there is no “save\_plot” argument
* added **GCMfuturestartYear** argument
* updated “readShapeLines” function to “readOGR”
* finds centroid of the user-defined bounding box in order to find closest GCM pixel rather than the old method of just choosing the next greatest GCM pixel in terms of lat and lon
* Leap year fix for GCMs that have leap days
* Fixed a Negative Precip Value filter that was being applied erroneously to temp and precip
* General internal data frame modifications (changed “Group.1” column name to “time\_year”, changed variables named “pr\_XXX” which represent all variables to “var\_XXX”
* Major changes to output graphics, now includes bounding boxes of region and GCM pixel demonstrated, all GCM pixels, and the second page has a period of record chart for all observed records of the given variable plotted

***Compare\_GCM:***

* Added SVD LSD Ocean Oscillation analysis
* Added **alignHistYears** argument to analyze common historic periods of observed and GCM data
* Cleaned up output graphics, so that graph legends dynamically change with series values, rather than cutting off values
* Improved flexibility for changing periods of analysis, so as to not have fixed starting and ending years, can change to any period of study without crashing, such as was the case with fixed loops of (i in 1:5) with arguments within like bYear = refDate[1] + (i – 1)\*30, which depended upon uniform base years and future time periods (i in 6:8) etc. which depended on having 150 years of historical data and 90 years of future, which limited the model data and study period you could use.

***knn\_bootrstrap***

* Added **JPmode** argument to choose between Yearly, Monthly, or Moving Window joint-probability schemes
* Added **alignHistYears** argument to allow for equivalent periods of historical GCM and observed periods for creating the joint probability matrices
* Added **HistRepro** argument to allow for production of synthetic series using just the observed historical joint probability matrices
* Added **extremesCorr** argument to allow for extremes correction within the function
* Fixed error in the original jp\_gcm\_avg variable assignment, which had been using the p95 value computed from the observed historical dataset, rather than the p95 from the GCM historical data
* Fixed major error in the assignment of the jp\_gcm\_f variable, which had been calculating the jp matrix using the historic GCM data frame instead of the future GCM data frame, in effect jp matrices were never being calculated with the future GCM ensemble data in the previous version, but only with historic GCM data. This command also had the same error of using the p95 value from the historical observed data. The values of jp\_gcm\_avg and jp\_gcm\_f were exactly the same, apart from the fact that the gcm\_ref\_idx indices used would be different, because the “future” one was using one year’s worth of values at a time instead of the whole time period, as was the historical.
* Removed the “if leap.year i in 0:365 vs I in 0:364” logic because the julian days only ever go up to 364, if it’s a leap year, both Feb-29 and Mar-1 have julian day values of 59, and therefore, they will be both included in the same windows of analysis, but it never reaches 365. February 29 is then excluded from the synthetic series to keep all years uniform at 365 days, and since the i loop only goes to 364
* Allowed easier modification of low threshold, now defined as “Threshold”, rather than needing to manually change each value in all commands
* Fixed many issues with dates, which occurred when GCMs had reference base years other than 1850, which had rendered some model data unusable before.
* Added extremes correction of time series, which directly corrects the output synthetic series, rather than just having the bucket of values having been corrected. Outputs final series as “MODEL\_ rcp45\_pr\_1\_bootstrap.csv/Rda”, and also if running in multi-region mode, exports the extremes correction key “MODEL\_bs\_pr\_rcp45\_day\_ExtremesCorrected\_key.Rda”
* Major upgrades to output graphics, many more additional graphics:
  + Exports graphics of the bootstrap boxplots which have monthly boxplots of mean daily and mean monthly values for (Observed vs. Synthetic, and Observed vs. GCM Hist vs. GCM Future vs. Synthetic, respectively)
  + Graphic of the average daily values of the upper and lower (dry/wet) thresholds of the Observed, GCM Hist, and GCM Future datasets.
  + Monthly plots of the tri-state transition matrix values for the historical observed, bootstrapped synthetic future series, GCM Hist, and GCM future datasets
  + Plots comparing the percentile values of the same 4 data series

***heatSignal***

* Added **maxGCMYear** argument
* Imports historical observed and GCM data for use in output graphics
* Fixes issue of blank values for the first and last 3 days of the heatSignal due to the weekly-averaging going on. Now just assigns those days as having the same value as the 4th and 4th-to-last days, respectively, so as to have a continuous time-series without 6 missing days for temperature.
* Simplified output table (removed junk columns of non-useful data)
* Added plot output of comparative boxplots (monthly avg) and histograms of temperature probability densities of the observed historic, GCM historic, and GCM future data sets

***sinteticSeries***

* Added **fullObsCatalog** argument when dealing with a very large basin, and want to apply the bootstrapped series obtained from one region to the whole catalog of station data in the broader basin
* Added **HistRepro** argument, for use in unison with the HistRepro argument in the knn\_bootstrap function
* Added **minRefYear** and **maxRefYear** arguments to deal with date uniformity
* Added function (when HistRepro = FALSE) which corrects the observed station data for all stations within a given region according to the extremes correction “key” or signature obtained for that region.
* Added large routine to deal with large multi-region basins, each of which has a separate extremes correction signature. It goes through each region, applies the appropriate extremes correction key, and stitches the extremes-corrected observed historical values together and exports them
* Removes leap-days from final series

***extremePR\_GPD***

* Added some arguments to make it more uniform with other functions (i.e. date arguments, model name, results folder etc.)
* Altered the routine to only find the extreme Threshold of the observed historic data, convert it to a percentile (i.e. p0.987), and then use that same percentile value for all the other extremes correction, rather than re-calculating a unique threshold for each GCM series (historic and future). This was done because having different percentile extreme thresholds for all the series resulted in non-uniform correction, with a “stretching” effect when one was being compared to another. For example, the curve representative of p0.985 and above for one time-series can’t be directly compared to that representing p0.995 and above of another. However, it was originally being done this way.
* Also modified the way in which the plots were generated, and the way in which the extremes were being corrected. Originally, a fairly opaque method of probabilistic sampling of certain extreme values were being done, though with the relatively small sample size of total number of extreme values (i.e. n = 100), it wasn’t guaranteed via this process that you would always capture the change in extremes being observed between the GCM hist and GCM future series. Therefore, now the sample size is artificially increased and uniformly applied to n = 10000 to ensure that the true shape of the extreme values curve is maintain when both graphed and when the correction is applied to produce the corrected observed series.
* Graphs generally improved to keep plots from running off the margins.