1. File structure:
   1. DeltaEstimator.r
      1. Purpose: main script that sets up the variables and arguments required for subsequent analysis. Will call upon other scripts to product results
      2. Files used: files that are used by this script during the analysis
         1. DriverDeltaSuisun.accdb
            1. Access database where all raw data is pulled from. RODBC driver to connect
         2. dm0.r
            1. Main analysis script. This script is what actually runs the asset evaluation and levee failure probability analysis
         3. doModel.r
            1. A wrapper script meant to tweak configurations for Upper Andrus and Brannan-Andrus islands. Uses dm0.r
         4. exec.ParRuns.r and do.Extract.r
            1. To run the analysis in parallel (multi-core) and to extract required data in a formatted manner. DID NOT explore these.
         5. Many different scripts in the bin folder. These are helper functions. They are described below
2. START:CHECK THESE THREE VARIABLES-CRUCIAL TO ENSURING ANALYSIS IS CORRECT
   1. Sets up crucial toggles for the scripts.
   2. What’s a base sheet?
3. Begin Initialization
   1. What’s bin code? What does it do?
      1. Bin code appears to be the helper functions
         1. Three scripts: Ancillary.r, Calculation.r, and Post\_AD\_Load.R
            1. The first two are loaded first and Post\_AD\_Load.R later
      2. Helper functions in Ancillary and Calculation:
         1. accumulate():
            1. Cumputes a running total of the input vector’s elements
         2. binaryMap():
            1. Converts a numeric vector into a binary vector, where elements less than or equal to 0 become 0, and elements greater than 0 become 1
         3. buildDataVecMIDs(): CHECK
            1. update the levee failure probability or flooding metrics for a specific island based on its flooding status and null stage recurrence domain information

For already flooded islands?

* + - 1. cdtlProbCoeffAdj():
         1. clips the product of y and alpha between 0 and 1
      2. checkLeveed():
         1. obtain the levee status for a specific island from a database and return it as a numeric value
      3. check0List():
         1. used to validate a list of elements based on specific criteria and can return different output formats based on the user's preference, such as the indices of valid elements, the island IDs of failed elements, or a comprehensive list containing both.
         2. Checks to see if a list of elements has a stage recurrence curve or not, i.e., if a certain island failed analysis or not.
      4. checkPackages():
         1. check if a list of specified packages are installed in the R environment, and if not, it installs and loads them
      5. extractionName():
         1. remove the 'AssetData.' prefix from a given string and replace underscores with spaces
      6. convertAssetValueToHeader():
         1. generate a formatted string based on the asset's information in a given asset code matrix and determine whether to add an enumeration or value suffix based on a damage curve index
      7. convertExclusionStatus():
         1. takes an island ID as input and returns a numeric value representing the island's status based on its flooding status, fragility curve, and error set
      8. convertGrowthFactor():
         1. replaces 2012 with 2010
      9. damageCurveIndex():
         1. takes an asset code as input and returns the corresponding asset damage curve code. If no damage curve code is found or it is 0, the function returns a default value of 11.
      10. doDateString():
          1. format the current system date in the day-month-year order with a custom separator specified by the datesep argument. If no separator is provided, the default separator is a dot ('.')
      11. fileBaseName():
          1. extract the base directory path from a given file name with its extension by finding the position of the last period and extracting the substring before it
      12. getTable():
          1. retrieve data from a specified table in a database or a data frame based on the provided column names and other parameters
      13. getWSDRSDat():
          1. retrieve WSDRData based on island name and scenario number
      14. pwlInterpolation():
          1. interpolate the probability value for a given input x based on a table of data with specified XcolumnHead and YcolumnHead, where the given x is used to filter the table and the associated value in Y (probability) is returned
      15. pwlPDFNormalize():
          1. returns the area under the curve (definite integral) associated with each value of x provided, returning the original x value and the definite integral. Uses the trapezoidal rule
      16. readRef():
          1. reads a table with a `dat` extension
      17. roundSpecial():
          1. applies different rounding techniques based on `type` specified between the values of x and r
      18. sheetCsvAdjust()
          1. checks if a given input string matches any patterns specified in a vector, and if not, it appends a file extension to the input string.
    1. Helper functions in Post\_AD\_Load.R
       1. convertScenario():
          1. retrieves a specific trait, based on `typeOut`, of a given scenario from the analysisData object
       2. getExternalTableData():
          1. filters a specific table in the refData object (analysisData) based on the given conditions in valList and extracts the values of a specified variable (extractVar)
       3. convertExternalData():
          1. A wrapper for getExternalTableData()

1. global variable initialization
   1. analysisData core list
      1. A list of different arguments used for the subsequent analysis
      2. Critical for the modeling that follows
2. Probability of levee failure: dm0.r script
   1. seismicFragility = a singular value that summarizes the probability of failure of a levee across a range of PGAs
      1. Take probability of failure given various PGA values associated with the levee
      2. Take probability of each PGA value occurring
      3. Dot product of these probability
      4. Sum products to calculate a representative value for the levee across entire PGA curve
         1. Value is constant across all scenarios (have not seen one that isn’t yet)
      5. Line 712
   2. Probability of levee failure = (inundationDepth) \* (probability of specific water level, stage)\*(probability of failure given a specific stage) + seismicFragility (calculated previously) – (probability of failure due to given stage)\*seismicFragility
      1. Failure can only occur for locations that can be flooded (InudationDepth, 0 or 1)
      2. The probability of a specific water level occurring (StageRecurrence)
      3. The probability of levee failure based on flooding (leveeFragility) + seismicFragility – chance of both occurring simultaneously
      4. Line 717 = net probability of failure across all the different stage levels
   3. Hydrologic levee failure: line 721
      1. Total summation of failure probability across all water stages
   4. Probability of levee failure = line 734