* The recent changes (Dec 2023) that I made to the code that calls getElevation() only works with GeoLight 2.0.1 because this new version of getElevation() has a method argument that the 2.0.0 version doesn’t, and the return value has changed. Need to test what version of GeoLight the user has and do the right thing.
* Update documentation for new features added to config file
* Switch to saving outputs from kernel contour to use sf instead of rgdal
* Switch to using sf as return type from do\_kernel
* Deal with warnings from st\_write:  
  Warning messages:  
  1: In CPL\_write\_ogr(obj, dsn, layer, driver, as.character(dataset\_options), :  
   GDAL Message 6: Field tFirst create as date field, though DateTime requested.  
  2: In CPL\_write\_ogr(obj, dsn, layer, driver, as.character(dataset\_options), :  
   GDAL Message 6: Field tSecond create as date field, though DateTime requested.
* Add new style config file to repo.
* Look into errors when doing kernel:  
  27: Could not parse expression: ‘`m`’. Returning as a single symbolic unit()  
  28: In sp::CRS(from$proj4string) : invalid PROJ4 string
* DONE (added file.opened()) R bombs if Excel has config file open when read\_cfg\_file tries to read it. Problem with readxl library? Figure out how to determine if the file is locked.
* DONE TODO: new version of getElevation returns a vector with the elevation angle  
  in second element. Need to check length of elev to figure out what to do.
* check on using pre- vs post- calibration data (different location)
* deal with

Warning messages:

1: In check\_tzones(e1, e2) : 'tzone' attributes are inconsistent

2: In check\_tzones(e1, e2) : 'tzone' attributes are inconsistent

at this line:

calib <- dplyr::filter(alldat, date >= as.POSIXct(cfg$calibStart, tz = "GMT") &

date <= as.POSIXct(cfg$calibEnd, tz = "GMT"))

probably need to explicitly set TZ for date to be GMT

* DONE add code to read activity data and return it.
* deal with multiple calibration data sets
* finish docs for all exported functions
* check all combinations of options:
  + keepCalibPoints
  + createShapeFile
* add ability to deal with multiple logger types (BAS, Migrate Tech, etc).

# Notes

* I briefly considered adding the sst correction stuff to this package (in branch sst\_correction) but changed my mind and put it in it’s own package instead (GLSSST)