

# tempdev20.R

sarah

2020-01-15

Converts a polygon boundary into a grid mesh, but in dataframe form

@param bdary List of points along the boundary of the simulation area (in lat/long) @param gridSize Grid size. Assumes square grid. In decimal form (eg 0.00025 = 25m<sup>2</sup> grid) @return A dataframe of lat/long coordinates representing the region described by bdary with space sizes gridSize.

```
make_grid <- function(bdary, gridSize){
  latval <- seq(min(bdary$lat), max(bdary$lat), gridSize)
  longval <- seq(min(bdary$long), max(bdary$long), gridSize)
  mesh <- expand.grid(latval, longval) #turns list of lat and long into grid
  points <- point.in.polygon(mesh$Var1, mesh$Var2, bdary$lat, bdary$long) #identifies grid points that
  grid <- cbind(mesh$Var1[which(points==1)], mesh$Var2[which(points==1)]) #puts these grid spaces into a
  grid <- as.data.frame(grid)

  return(grid)
}
```

Determines the grid ID number for each grid space within a bounding box. Useful for, eg, large areas of greenery that we can easily draw a boundary around, instead of grabbing the lat/long of every point within the region. Returns a dataframe of grid ID numbers that will be associated with that land type.

@param bb Bounding box of region, in lat/long @param gridSize Grid size. Assumes square grid. In decimal form (eg 0.00025 = 25m<sup>2</sup> grid) @param grid Dataframe of area grid, as per what's created in make\_grid @return A dataframe of grid IDs associated with that land type.

```
bb_to_matchedIDs <- function(bb, gridSize, grid){
  # Passing "grid" might be slow, but let's just do it for now so I can get this code done
  mesh <- expand.grid(seq(min(bb$lat), max(bb$lat), gridSize), seq(min(bb$long), max(bb$long), gridSize))
  points <- point.in.polygon(mesh$Var1, mesh$Var2, bb$lat, bb$long) #identifies grid points that lie wi

  bb.grid <- cbind(mesh$Var1[which(points > 0)], mesh$Var2[which(points > 0)]) #puts these grid spaces
  bb.grid <- unique(as.data.frame(bb.grid)) # Get rid of duplicates

  matched.IDs <- lapply(1:nrow(bb.grid), function(x){assign_grid(bb.grid[x, ], grid)}) #the 'grid' is o
  grid.pos <- unlist(matched.IDs)
  matched.IDs <- as.data.frame(grid.pos)
  return(matched.IDs)
}
```

Determines the grid ID number corresponding to each lat/long point in the input. Returns a dataframe of grid ID numbers that will be associated with that land type.

@param points Lat/long points to be matched to a grid ID. @param grid Dataframe of area grid, as per what's created in make\_grid @return A dataframe of grid IDs associated with that land type.

```
points_to_matchedIDs <- function(points, grid){
  IDs <- lapply(1:nrow(points), function(x){assign_grid(points[x, ], grid)})
  grid.pos <- unlist(IDs)
  matched.IDs <- as.data.frame(grid.pos)
  return(grid.pos)
}
```

```
}
```

Assign a grid ID to a particular lat/long point. Used for functions `points_to_matchedIDs` and `bb_to_matchedIDs`. @param `fake_BOMdata` A test set of temperatures. @param `noDays` Number of days in the simulation. @param `tempdata` Dataframe which includes the temp. deviate for each grid position @return A dataframe of each grid position, its temperature deviate, and localised temperatures in that grid space over time.

```
assign_grid <- function(pos, grid){
  close <- which((abs(pos$V2 - grid$V2) < gridSize) & abs(pos$V1 - grid$V1 < gridSize))
  grid.match <- -1
  if(length(close > 0)){
    close.dist <- list(abs(pos$V2 - grid$V2[close]), abs(pos$V1 - grid$V1[close]))
    closest <- which(close.dist[[1]] == min(close.dist[[1]]) & close.dist[[2]] == min(close.dist[[2]]))
    grid.match <- close[closest]
  }
  else if(length(close == 0)){
    grid.match <- -1
  }
  return(grid.match)
}
```