Zone Assignment

Bryce McManus

2022-02-17

Contents

Intro

This document shows how observations are assigned in FishSET and compares zone assignments with those found in the most recent version of the scallop data (provided Oct/Nov 2021). It then illustrates certain factors that can affect zone assignment: polygon boundary types, spherical v. planar geometry, and spatial joins.

Library

```
# Set Path
here::i_am("scallop_zone_assignment.Rmd")
```

```
## here() starts at /net/home2/mlee/Effort-Displacement---Scallop
```

```
# Install Packages if necessary and Load Libraries
PKG <- c("here","leaflet", "tidyverse", "sf")
for (p in PKG) {
  if(!require(p,character.only = TRUE)) {
    install.packages(p)
    require(p,character.only = TRUE)}
}
```

```
## Loading required package: here
```

```
## Loading required package: leaflet
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.7.2, GDAL 3.1.3, PROJ 6.3.2
```

```
# Set data vintage
vintage_string<-"2022_02_16"
```

Data

This is the data from the most recent Kiteworks folder that was shared with the FishSET team in Oct/Nov of last year.

```
# load in Oct 18 data.
scallop_10_18 <- read_csv(here("data","main","cost_join_10_18.csv"))
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 34737 Columns: 35
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (7): VTR_PORT, VTR_STATE, GEARCODE, SECGEARFISH, SPPNAME, OPERATOR, NAME
## dbl  (26): ...1, TRIPID, IMGID, YEAR, TRIP_LENGTH, DEALNUM, DOLLAR, POUNDS, ...
## date  (1): Date
## time  (1): Time
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# load in final product
final_product_savename<-paste0("final_product_lease",vintage_string,".Rds")
final_product_lease<-readRDS(here("data","main",final_product_savename))

# overwrite
#scallop_10_18<-final_product_lease

# Load in 10 minute squares
tenMinSqr_new <-
  here("data",
       "external",
       "shapefiles",
       "Ten Minute Squares Cut North and Greater Atlantic") %>%
  st_read() %>%
  st_zm() # remove Z/M dimensions from feature
```

```
## Reading layer `Ten_Minute_Squares_Clip6' from data source
##   `/net/home2/mlee/Effort-Displacement---Scallop/data/external/shapefiles/Ten Minute Squares Cut Nor
##   using driver `ESRI Shapefile'
## Simple feature collection with 2410 features and 15 fields
## Geometry type: POLYGON
## Dimension:     XYZ
```

```
## Bounding box:  xmin: -77.33333 ymin: 35.33333 xmax: -65 ymax: 45.33333
## z_range:       zmin: 0 zmax: 0
## Geodetic CRS:  NAD83
```

Check for duplicate trip IDs.

```
nrow(scallop_10_18)
```

```
## [1] 34737
```

```
scallop_10_18 %>%
  distinct(TRIPID) %>%
  nrow()
```

```
## [1] 34737
```

Zone assignment

This is the approach we take in FishSET to assign observations to zones.

```
# create sf version of data, convert to WGS84
# crs <- st_crs(tenMinSqr_new) # using crs from ten minute squares didn't change results
crs <- 4326

scallop_sf <-
  st_as_sf(x = scallop_10_18, coords = c("DDLON", "DDLAT"),
           crs = crs)

# convert Squares to WGS84
tenMinSqr_new <- st_transform(tenMinSqr_new, crs = st_crs(scallop_sf))

# same results as st_within
inter <- sf::st_intersects(scallop_sf, tenMinSqr_new)

inter_save <- inter

 if (any(lengths(inter) > 1)) { # if more than one zone intersects, assign to closest zone

  dub <- which(lengths(inter) > 1)
  inter[dub] <- st_nearest_feature(scallop_sf[dub,], tenMinSqr_new)
 }

# Add ZoneID column to data
pts <- as.data.frame(as.numeric(inter))
colnames(pts) <- "col.id"
pts$ID <- tenMinSqr_new$MN10SQID[pts$col.id]
scallop_sf$ZoneID <- pts$ID
```

52 points did not intersect a zone and 195 intersected multiple zones. The rest intersected with one zone. Non-intersecting points will show up as `NA` in the `ZoneID` column.

```
table(lengths(inter_save))
```

```
##
##     1
## 34737
```

After reassigning zones with 2+ intersections to the nearest zone.

```
table(lengths(inter))
```

```
##
##     1
## 34737
```

FishSET v. NE zone assignment

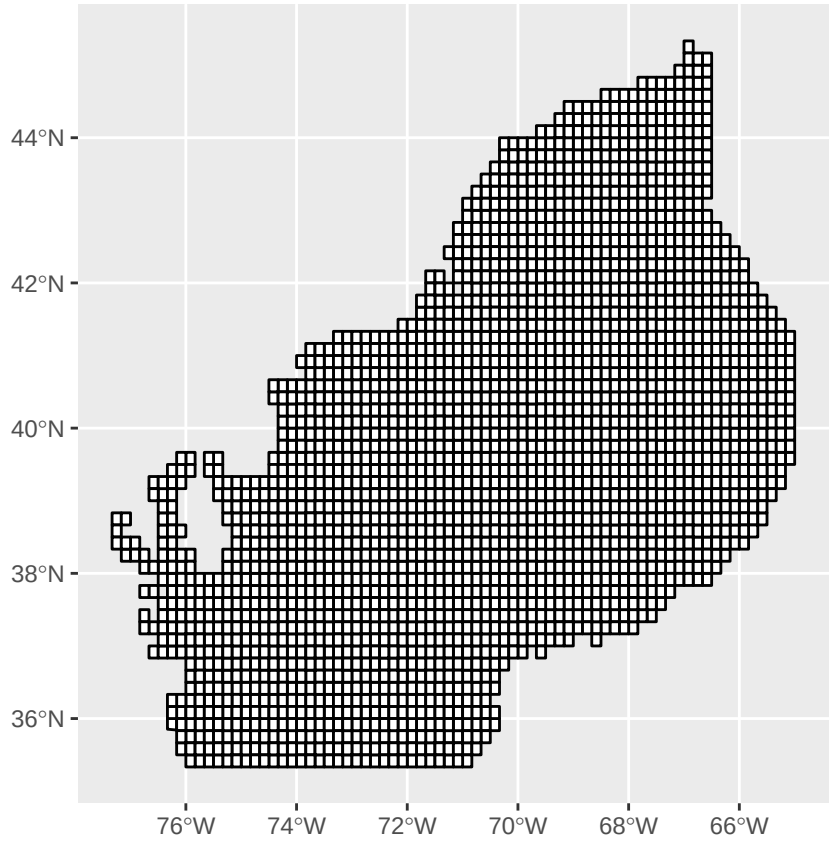
Just to recap: `ZoneID` is the zone assignment column created using the FishSET approach. `MN10SQID` is the zone ID column that came with the updated scallop data. There is a 6.15% difference (2086 obs) in zone assignments between `ZoneID` and `MN10SQID`.

```
scallop_sf <- scallop_sf %>%
  mutate(fs_ne_match = ZoneID == MN10SQID)

# add ZoneID and match col to scallop_10_18
scallop_10_18 <-
  scallop_sf %>%
  st_set_geometry(NULL) %>%
  select(ZoneID, fs_ne_match) %>%
  bind_cols(scallop_10_18, .)

# summary table of matching IDs
scallop_sf %>%
  st_set_geometry(NULL) %>%
  summarize(n_obs = n(),
            id_match = sum(fs_ne_match, na.rm = TRUE),
            id_no_match = sum(!fs_ne_match, na.rm = TRUE),
            perc_match = id_match/n() * 100,
            perc_no_match = 100 - perc_match)
```

```
## # A tibble: 1 x 5
##   n_obs id_match id_no_match perc_match perc_no_match
##   <int>    <int>       <int>      <dbl>         <dbl>
## 1 34737    33523        1214       96.5          3.49
```

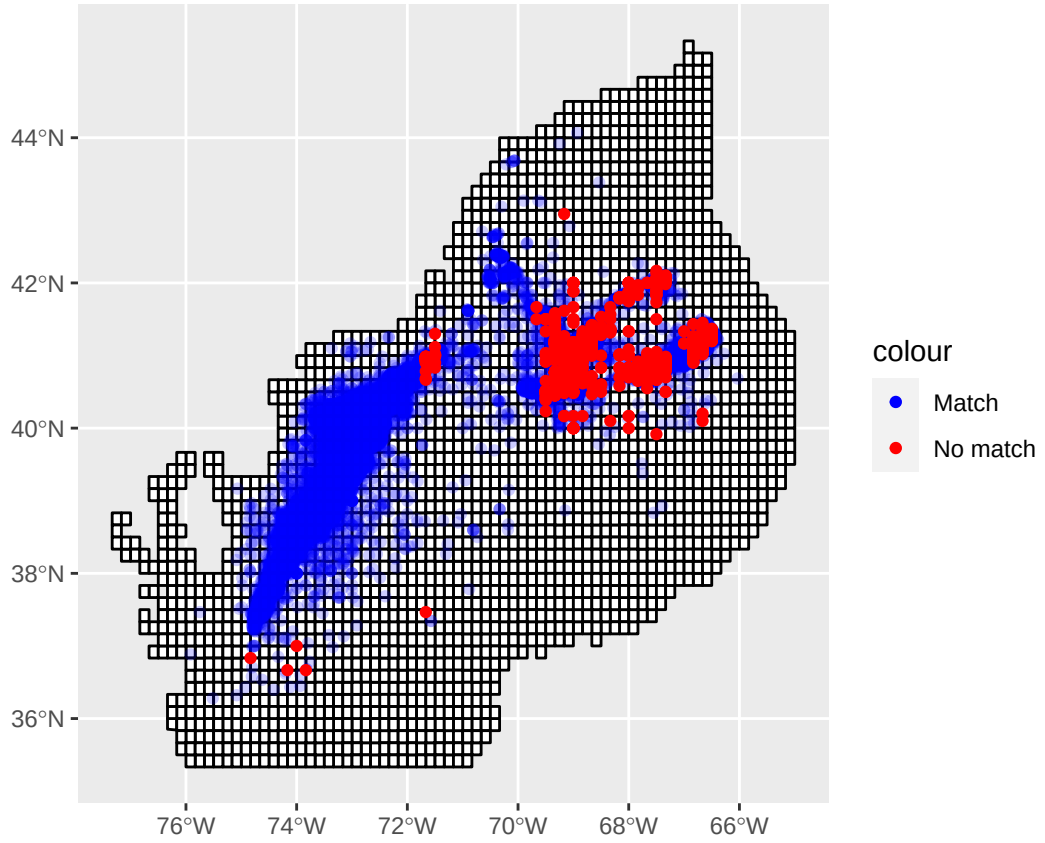
All the unassigned points in `ZoneID` fall on a zone intersection.

```
s_bbox <- st_bbox(filter(scallop_sf, is.na(ZoneID)))

ggplot() +
  geom_sf(data = tenMinSqr_new, fill = "white", color = "black") +
  geom_sf(data = filter(scallop_sf, is.na(ZoneID)),color = "red") +
  coord_sf(xlim = c(s_bbox[1], s_bbox[3]), ylim = c(s_bbox[2], s_bbox[4]))
```



The difference in zone assignment appears only to occur on zone boundaries and intersections.

```
ggplot() +
  geom_sf(data = tenMinSqr_new, fill = "white", color = "black") +
  geom_sf(data = filter(scallop_sf, fs_ne_match),
          aes(color = "Match"),
          alpha = .2,
          show.legend = "point") +
  geom_sf(data = filter(scallop_sf, !fs_ne_match),
          aes(color = "No match"),
          show.legend = "point") +
  coord_sf(xlim = c(s_bbox[1], s_bbox[3]), ylim = c(s_bbox[2], s_bbox[4])) +
 scale_color_manual(values = c("Match" = "blue", "No match" = "red"))
```



This leaflet plot shows the first 20 non-matching zone assignments. Each point has a label showing the zone assigned by FishSET and by NE.

```
# marker labels for leaflet plots
scallop_10_18 <-
  scallop_10_18 %>%
  mutate(ZoneID_lab = paste0("FS: ", ZoneID, "<br/>"),
         MN10SQID_lab = paste0("NE: ", MN10SQID, "<br/>")) %>%
  unite("unite_lab", ZoneID_lab, MN10SQID_lab, sep = " ") %>%
  mutate(unite_lab = paste0(unite_lab, "lon: ", DDLON, "<br/>lat: ", DDLAT))

fs_no_match <- scallop_10_18 %>%
  filter(fs_ne_match == FALSE) %>%
  head(20) %>%
  pull(ZoneID)

ne_no_match <- scallop_10_18 %>%
  filter(fs_ne_match == FALSE) %>%
  head(20) %>%
  pull(MN10SQID)

leaflet() %>%
  addTiles() %>%
  addPolygons(data = filter(tenMinSqr_new, MN10SQID %in% c(fs_no_match, ne_no_match)),
              fillColor = "white",
              fillOpacity = 0.5,
              color = "black",
```

```
              stroke = TRUE,
              weight = 1,
              labelOptions = labelOptions(noHide = TRUE),
              layerId = ~MN10SQID,
              label = ~MN10SQID) %>%
  addMarkers(data = head(filter(scallop_10_18, fs_ne_match == FALSE), 20),
             lng = ~DDLON, lat = ~DDLAT,
             label = ~lapply(unite_lab, htmltools::HTML))
```

Similar to the plot above, but comparing the unassigned points in `ZoneID` with `MN10SQID`.

```
if (sum(is.na(scallop_10_18$ZoneID)) > 0) {

  fs_na <- scallop_10_18 %>%
  filter(is.na(ZoneID)) %>%
  pull(MN10SQID)

leaflet() %>%
  addTiles() %>%
  addPolygons(data = filter(tenMinSqr_new, MN10SQID %in% fs_na),
              fillColor = "white",
              fillOpacity = 0.5,
              color = "black",
              stroke = TRUE,
              weight = 1,
              labelOptions = labelOptions(noHide = TRUE),
              layerId = ~MN10SQID,
              label = ~MN10SQID) %>%
  addMarkers(data = filter(scallop_10_18, is.na(ZoneID)),
             lng = ~DDLON, lat = ~DDLAT,
             label = ~lapply(unite_lab, htmltools::HTML))
}
```

Factors that may affect point assignment

Polygon boundary types

Point assignment can vary depending on whether polygon boundaries are treated as "closed", "open", or "semi-open".

```
inter_closed <- lengths(st_intersects(scallop_sf, tenMinSqr_new, model = "closed"))
inter_open <- lengths(st_intersects(scallop_sf, tenMinSqr_new, model = "open"))
inter_semi <- lengths(st_intersects(scallop_sf, tenMinSqr_new, model = "semi-open"))

bound_type <- tibble(Closed = inter_closed, Open = inter_open, Semi = inter_semi)

pivot_longer(bound_type, cols = c("Closed", "Open", "Semi"),
             names_to = "boundary_type", values_to = "Zones_assigned") %>%
  group_by(boundary_type) %>%
  count(Zones_assigned)
```

```
## # A tibble: 3 x 3
## # Groups:   boundary_type [3]
##   boundary_type Zones_assigned     n
```

```
##   <chr>                  <int> <int>
## 1 Closed                     1 34737
## 2 Open                       1 34737
## 3 Semi                       1 34737
```

sf version < 1.0

Sf 1.0+ uses spherical geometry (S2) on coordinates, sf < 1.0 uses planar geometry (R2).

```
# test difference in S2 and R2 methods
sf_use_s2(FALSE)
```

```
## Spherical geometry (s2) switched off
```

```
inter_closed_r2 <- lengths(st_intersects(scallop_sf, tenMinSqr_new, model = "closed"))
```

```
## although coordinates are longitude/latitude, st_intersects assumes that they are planar
## although coordinates are longitude/latitude, st_intersects assumes that they are planar
```

```
inter_open_r2 <- lengths(st_intersects(scallop_sf, tenMinSqr_new, model = "open"))
```

```
## although coordinates are longitude/latitude, st_intersects assumes that they are planar
## although coordinates are longitude/latitude, st_intersects assumes that they are planar
```

```
inter_semi_r2 <- lengths(st_intersects(scallop_sf, tenMinSqr_new, model = "semi-open"))
```

```
## although coordinates are longitude/latitude, st_intersects assumes that they are planar
## although coordinates are longitude/latitude, st_intersects assumes that they are planar
```

```
sf_use_s2(TRUE)
```

```
## Spherical geometry (s2) switched on
```

```
bound_type_r2 <- tibble(Closed = inter_closed_r2,
                        Open = inter_open_r2,
                        Semi = inter_semi_r2)

pivot_longer(bound_type_r2, cols = c("Closed", "Open", "Semi"),
             names_to = "boundary_type", values_to = "Zones_assigned") %>%
  group_by(boundary_type) %>%
  count(Zones_assigned)
```

```
## # A tibble: 3 x 3
## # Groups:   boundary_type [3]
##   boundary_type Zones_assigned     n
##   <chr>                  <int> <int>
## 1 Closed                     1 34737
## 2 Open                       1 34737
## 3 Semi                       1 34737
```

Spatial join

Spatial joins will add additional rows if a point intersects more than one zone. If an inner join is used (`left = FALSE`), points that don't intersect zones are dropped.

```
scallop_sf_join <- st_join(scallop_sf,
                           tenMinSqr_new["MN10SQID"],
                           suffix = c(".x", ".y"),
```

```
                           left = TRUE)

scallop_sf_join <-
  scallop_sf_join %>%
  mutate(fs_join_match = ZoneID == MN10SQID.y,
         ne_join_match = MN10SQID.x == MN10SQID.y)

scallop_sf_join %>%
  st_set_geometry(NULL) %>%
  summarize(n_obs = n(),
    fs_join_match = sum(fs_join_match, na.rm = TRUE),
    ne_join_match = sum(ne_join_match, na.rm = TRUE),
    fs_ne_match = sum(fs_ne_match, na.rm = TRUE))
```

```
## # A tibble: 1 x 4
##   n_obs fs_join_match ne_join_match fs_ne_match
##   <int>         <int>         <int>       <int>
## 1 34737         34737         33523       33523
```

Spatial join adds 195 rows: one for each point that intersected two zones.

```
nrow(scallop_sf_join)
```

```
## [1] 34737
```

```
scallop_sf_join %>%
  st_set_geometry(NULL) %>%
  distinct(TRIPID) %>%
  nrow()
```

```
## [1] 34737
```

Table of matching zones by FishSET, NE, and spatial join approaches.

```
scallop_sf_join %>%
    st_set_geometry(NULL) %>%
    count(fs_ne_match, fs_join_match, ne_join_match)
```

```
## # A tibble: 2 x 4
##   fs_ne_match fs_join_match ne_join_match     n
##   <lgl>       <lgl>         <lgl>         <int>
## 1 FALSE       TRUE          FALSE          1214
## 2 TRUE        TRUE          TRUE          33523
```

All 195 of the cases from the first row of the table above (no shared zone assignments) are from points assigned to multiple zones. No approach handled this case in the same way.

```
dup_trip_id <- scallop_10_18$TRIPID[lengths(inter_save) > 1]

scallop_sf_join <-
  scallop_sf_join %>%
  mutate(dup_zone = TRIPID %in% dup_trip_id)

scallop_sf_join %>%
  st_set_geometry(NULL) %>%
  filter(!fs_ne_match & !fs_join_match & !ne_join_match) %>%
  summarize(dup_zone = sum(dup_zone))
```

```
## # A tibble: 1 x 1
##   dup_zone
##      <int>
## 1        0
```